Calhoun: Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1972-06

A study of supersonic cascade flutter

Chalkley, Henry George

http://hdl.handle.net/10945/16389

Downloaded from NPS Archive: Calhoun

DUDLEY KNOX LIBRARY

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

A STUDY OF SUPERSONIC
CASCADE FLUTTER

Henry George Chalkley

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Study of Supersonic
Cascade Flutter

by

Henry George Chalkley

Thesis Advisor: M. F. Platzer

June 1972

Approved for public release; distribution unlimited.

A Study of Supersonic
Cascade Flutter

by

Henry George Chalkley
Lieutenant, United States Navy
M.S., Naval Postgraduate School, 1971

Submitted in partial fulfillment of the
requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1972

Thesis
C33865
c.1

ABSTRACT

Supersonic flow past oscillating flat plate cascades with supersonic leading-edge locus is analysed using a linearized method of characteristics valid for arbitrary frequencies and an elementary analytical theory valid only for low frequencies of oscillation. These two methods are extensions of previous work by Teipel and Sauer for the single airfoil in an unbounded supersonic flow to the case of airfoils oscillating in cascade. Included is the determination of pressure distribution and both a two-degree-of-freedom (bending and torsion) flutter analysis and a single-degree-of-freedom (torsion) flutter analysis. Numerically determined flutter boundaries are presented for various primary parameters such as, Mach number, solidity, stagger angle, density ratio, structural damping coefficient, and elastic axis position. Also, results are presented for the related problem of supersonic wind tunnel interference (including the effect of tunnel porosity) and airfoil-airfoil interference.

TABLE OF CONTENTS

I. INTRODUCTION ---------------------------------- 8

II. PROBLEM FORMULATION ----------------------------11

III. METHOD OF CHARACTERISTICS ---------------------19

IV. ELEMENTARY THEORY FOR SLOWLY OSCILLATING CASCADES ----------------------------38

V. FLUTTER ANALYSIS --------------------------------49

VI. COMPUTATIONAL PROCEDURE ------------------------59

VII. RESULTS --69

- A. COMPARISON OF ELEMENTARY THEORY WITH MILES ----------------------------------69
- B. COMPARISON OF ELEMENTARY THEORY WITH LANE -----------------------------------71
- C. COMPARISON OF ELEMENTARY THEORY WITH DRAKE ----------------------------------71
- D. AIRFOIL-AIRFOIL INTERFERENCE ------------73
- E. COMPARISON OF THE METHOD OF CHARACTERISTICS WITH ELEMENTARY THEORY -------------75
- F. COMPARISON OF THE METHOD OF CHARACTERISTICS WITH GARRICK-RUBINOW ---------------79
- G. METHOD OF CHARACTERISTICS RESULTS -------79

VIII. CONCLUSIONS AND RECOMMENDATIONS -------------88

APPENDIX A - COMPUTER PROGRAM VARIABLES ------------90

APPENDIX B - FLOW DIAGRAM - PROGRAM A --------------97

COMPUTER PROGRAM A ---------------------------------110

COMPUTER PROGRAM B ---------------------------------133

LIST OF REFERENCES ---------------------------------153

INITIAL DISTRIBUTION LIST --------------------------155

FORM DD 1473 ---------------------------------------156

TABLE OF SYMBOLS

A	$d \cot \alpha$
B	stagger, $d \tan \beta$
C	Teipel complex velocity of sound perturbation amplitude
c	velocity of sound perturbation
$\bar{c}$	local velocity of sound
c_∞	freestream velocity of sound
$\hat{c}$	blade chord
$\cot \alpha$	$\sqrt{M^2-1}$
C_m	pitching moment coefficient
$c_{m\theta}$	pitching moment coefficient amplitude (real)
$c_{m\dot{\theta}}$	pitching moment coefficient amplitude (imaginary) or pitch damping coefficient
C_p	pressure coefficient amplitude
C_h	blade bending stiffness
C_θ	blade torsional stiffness
d	blade distance
g	structural damping
h	blade plunging oscillation
h_0	blade plunging oscillation amplitude
I_θ	blade moment of inertia
k	reduced frequency $(\frac{\omega\hat{c}}{u_\infty})$
L	aerodynamic lift
M	Mach number
M_b	blade mass per unit span
M_θ	aerodynamic pitching moment

$\mathcal{M}$	molecular weight
P	non-dimensional complex pressure perturbation amplitude
p	pressure
p_∞	freestream pressure
r_θ	blade radius of gyration $(\sqrt{\frac{4I_\theta}{M_b \hat{c}^2}})$
$\mathcal{R}$	universal gas constant
s	entropy
S_θ	blade static moment about the elastic axis per unit span
T	temperature
U	Teipel complex streamwise velocity perturbation amplitude
U_F	freestream flutter speed
u	streamwise velocity perturbation
u	internal energy in Eq.(59)
$\bar{u}$	local streamwise velocity
u_∞	freestream velocity
V	Teipel complex normal velocity perturbation amplitude
v	normal velocity perturbation
X	frequency rate $[(\frac{\omega_\theta}{\omega})^2]$
x_θ	distance between blade center of gravity and elastic axis $(\frac{S_\theta}{M_b})$
x_0	elastic axis position
x,y	Cartesian coordinates
α	Mach angle
β	stagger angle (complimentary)

γ	ratio of specific heats
δ	interblade phase angle
Φ	complex perturbation velocity potential
ϕ	complex perturbation velocity potential amplitude
θ	angle of attack
θ_o	amplitude of pitch oscillation
μ	blade density-parameter $(\frac{M_b}{\rho_\infty \hat{c}^2})$
ω	frequency of oscillation
ω_F	flutter frequency
ω_θ	torsional natural frequency $(\sqrt{\frac{C_\theta}{I_\theta}})$
ω_h	bending natural frequency $(\sqrt{\frac{C_h}{M_b}})$
Ω_θ	factor (μr_θ^2)
Ω	factor $[\mu(\frac{\omega_h}{\omega_\theta})^2]$
ρ	local density
ρ_∞	freestream density
ξ,η	characteristic coordinates
σ	wind tunnel porosity
χ	defined in Eq.(118)
Ψ	defined in Eq.(118)
ν	grid fineness ratio

ACKNOWLEDGMENT

This thesis was written with the close support and continual assistance of Associate Professor M. F. Platzer.

I. INTRODUCTION

In order to design an effective supersonic compressor, an analysis of the complex unsteady flow phenomena in this speed regime is necessary. Since the infinite cascade has long been used by the engineer to model two-dimensional compressor flows, the study of oscillating supersonic cascades is important in predicting the flutter characteristics or dynamic response of these compressors. Moreover, the basic understanding of the flow gained by such a study can lead to better design criteria.

Two basic cascade configurations can be distinguished at the outset; i.e., cascades with either subsonic or supersonic leading-edge locus. Although the former is the case of primary interest in current research, in this paper the case of supersonic leading-edge locus is treated in order to form a basis for extension to the more complicated case of subsonic leading-edge locus. This method of approach also has the advantage that the problem of oscillatory wind tunnel interference, already analysed in previous work, is contained in the theory as a special case.

The interference problem of linearized supersonic flow past airfoils oscillating between solid wind tunnel walls was considered by Miles (1956) who derived a solution using Laplace transform techniques. Drake (1956) treated this case for wind tunnels with free jet boundaries and later Drake (1957) gave a solution to this interference problem for

porous-wall wind tunnels also using Laplace transform methods. In an extension of Miles' work, Lane (1957) presented a solution for supersonic flow past oscillating cascades with supersonic leading-edge locus again using Laplace transform techniques. Further work in this area was done by Hamamoto (1962). Using Teipel's (1962) method of characteristics approach, Platzer and Pierce (1970) made an analysis of oscillatory supersonic wind tunnel interference and with the help of the high speed computer were able to predict the pressure distribution along an airfoil oscillating with arbitrary frequency in solid, free jet, or porous-wall wind tunnels. Platzer (1971) will present an elementary analysis of porous-wall wind tunnel interference effects which generalizes Sauer's (1950) solution for a single airfoil oscillating at low frequency in an unbounded linearized supersonic flow. Platzer and Chalkley (1972) further extended this solution to form an elementary theory that together with a method of characteristics procedure (based on Platzer and Pierce, 1970) is used to analyse the supersonic flow past oscillating flat plate cascades having supersonic axial velocities but otherwise arbitrary stagger angle.

In this thesis the detailed description of both the elementary theory and the method of characteristics that was not possible in Platzer and Chalkley (1972) is given. With the aid of the high speed computer, the method of characteristics procedure is used to predict the pressure distributions and aerodynamic forces and pitching moments on a typical cascade

blade. Both a two-degree-of-freedom (torsion and bending) flutter analysis and a one-degree-of-freedom (torsion) flutter analysis are presented, with numerical results given for the case of the torsional flutter. The elementary theory (valid only for low frequencies of oscillation) is used to provide a convenient check of the characteristics approach as well as to easily predict a number of dynamic instability boundaries for a slowly oscillating airfoil subjected to interference from supersonic wind tunnel walls in one case and from a larger airfoil in close proximity in yet another case.

II. PROBLEM FORMULATION

The cascade is considered to be one of two-dimensional flat plates with each blade performing equal low amplitude, simple harmonic oscillations of the same mode with the same interblade phase angle. As shown in Figure 1, it is aligned in the x-y coordinate system such that one blade lies along the x-axis with its leading-edge at the origin.

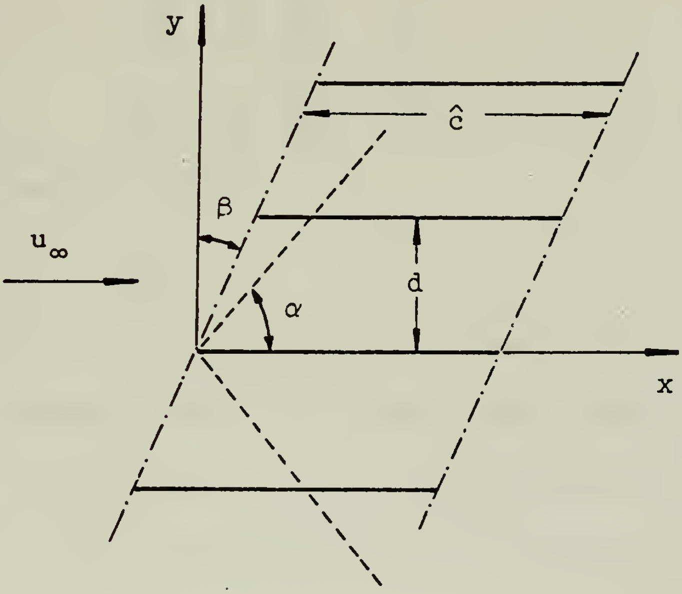


Figure 1

The cascade is further considered to have supersonic leading-edge locus and, except for this restriction, arbitrary stagger, such that,

$$\tan \beta \leq \cot \alpha \tag{1}$$

With this assumption, the flow between any two adjacent blades can be used to describe the flow through the entire cascade. Since disturbances cannot travel upstream of the

Mach lines from the leading-edge of a blade, interference from other than adjacent blades is not possible in the supersonic flow between them. Hence, only the flow between the blade along the x-axis and the one above it is considered. This flow is assumed to be the non-viscous flow of a perfect gas governed by the continuity equation,

$$\frac{D\rho}{Dt} + \rho \left(\frac{\partial \bar{u}}{\partial x} + \frac{\partial v}{\partial y}\right) = 0 \tag{2}$$

the Euler equations,

$$\frac{D\bar{u}}{Dt} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0 \tag{3a}$$

$$\frac{Dv}{Dt} + \frac{1}{\rho}\frac{\partial p}{\partial y} = 0 \tag{3b}$$

and the energy equation,

$$\frac{Ds}{Dt} = 0 \tag{4}$$

Because of the assumption of small amplitude oscillations of the cascade blades, all flow quantities are considered to be small perturbations linearly superimposed on freestream quantities. The velocities are written as

$$\bar{u} = u_\infty + u \tag{5a}$$

$$\bar{c} = c_\infty + c \tag{5b}$$

while the pressure and density perturbations are written as

$$\Delta p = p - p_\infty \tag{6a}$$

$$\Delta \rho = \rho - \rho_\infty \tag{6b}$$

With the assumption of isentropic flow, the local velocity of sound can be defined as,

$$\bar{c}^2 = \frac{dp}{d\rho} \tag{7}$$

further,

$$\frac{p}{\rho^{\gamma}} = \text{constant} \tag{8}$$

Taking the total differential,

$$\frac{1}{\rho^{\gamma}}\,dp - \gamma p\,\frac{1}{\rho^{\gamma+1}}\,d\rho = 0 \tag{9}$$

or

$$\frac{dp}{d\rho} = \gamma\,\frac{p}{\rho} \tag{10}$$

hence,

$$\bar{c}^2 = \gamma\,\frac{p}{\rho} \tag{11}$$

The flow boundary condition along the blade surfaces [as shown in Ch. 5, Bisplinghoff, Ashley, Halfman (1955) for a simple airfoil] is as follows: If the surface over which the fluid flows has the equation,

$$F\,(x,\,y,\,t) = 0 \tag{12}$$

particles in contact with the surface must have the same normal velocity as the surface. Stated differently, the rate of change of F is zero when the motion of a particular fluid element is followed along the surface, or

$$\frac{DF}{Dt} = 0 \tag{13}$$

If an arbitrary, thin, two-dimensional cascade blade is considered, the equation of the upper surface can be written,

$$F_u(x,y,t) = y - y_u\,(x,t) = 0 \tag{14}$$

where $y = 0$ is the mean camber line and y_u is the distance from the mean camber line to the upper surface. Likewise, the equation of the lower surface can be written,

$$F_L\ (x,y,t) = y - y_L\ (x,t) = 0 \qquad (15)$$

At $y = y_u$

$$\frac{DF_u}{Dt} = -\frac{\partial y_u}{\partial t} - \bar{u}\frac{\partial y_u}{\partial x} + v = 0 \qquad (16)$$

and at $y = y_L$

$$\frac{DF_L}{Dt} = -\frac{\partial y_L}{\partial t} - \bar{u}\frac{\partial y_L}{\partial x} + v = 0 \qquad (17)$$

since

$$\frac{\partial y_u}{\partial y} = \frac{\partial y_L}{\partial y} = 1 \qquad (18)$$

Thus, the normal flow velocity can be written,

$$v = \frac{\partial y_u}{\partial t} + \bar{u}\frac{\partial y_u}{\partial x} \qquad \text{at} \quad y = y_u(x,t) \qquad (19a)$$

$$v = \frac{\partial y_L}{\partial t} + \bar{u}\frac{\partial y_L}{\partial x} \qquad \text{at} \quad y = y_L(x,t) \qquad (19b)$$

By applying the assumption of linear perturbations on the freestream, the normal velocity equations become

$$v = \frac{\partial y_u}{\partial t} + u_\infty\frac{\partial y_u}{\partial x} + u\frac{\partial y_u}{\partial x} \qquad \text{at } y = y_u(x,t) \qquad (20a)$$

$$v = \frac{\partial y_L}{\partial t} + u_\infty\frac{\partial y_L}{\partial x} + u\frac{\partial y_L}{\partial x} \qquad \text{at } y = y_L(x,t) \qquad (20b)$$

Since a thin blade is being considered y_u and y_L are small, thus the equations become, cancelling the higher order terms,

$$v = \frac{\partial y_u}{\partial t} + u_\infty\frac{\partial y_u}{\partial x} \qquad \text{at } y = y_u(x,t) \qquad (21a)$$

$$v = \frac{\partial y_L}{\partial t} + u_\infty\frac{\partial y_L}{\partial t} \qquad \text{at } y = y_L(x,t) \qquad (21b)$$

This normal velocity can be further written as a Taylor series expansion of the normal flow velocity at $y = 0$. Thus,

$$v(x,y_u,t) = v(x,0^+,t) + y_u \frac{\partial v(x,0^+,t)}{\partial y} + \frac{y_u^2}{2!} \frac{\partial^2 v(x,0^+,t)}{\partial y^2} + \ldots\ldots \tag{22a}$$

and

$$v(x,y_L,t) = v(x,0^-,t) + y_L \frac{\partial v(x,0^-,t)}{\partial y} + \frac{y_L^2}{2!} \frac{\partial^2 v(x,0^-,t)}{\partial y^2} + \ldots\ldots \tag{22b}$$

Again with the assumption of a thin blade and small linear perturbations on the freestream: y_u, y_L, and v are small and higher order terms may again be cancelled from the equations leaving:

$$v(x,y_u,t) = v(x,0^+,t) \tag{23a}$$

$$v(x,y_L,t) = v(x,0^-,t) \tag{23b}$$

Thus,

$$v(x,y,t) = \frac{\partial y_u}{\partial t} + u_\infty \frac{\partial y_u}{\partial x} \qquad \text{at } y = 0^+ \tag{24a}$$

$$v(x,y,t) = \frac{\partial y_L}{\partial t} + u_\infty \frac{\partial y_L}{\partial x} \qquad \text{at } y = 0^- \tag{24b}$$

With the assumption of simple harmonic motion of the blade,

$$y_u(x,t) = y_u(x)\, e^{i\omega t} \tag{25a}$$

$$y_L(x,t) = y_L(x)\, e^{i\omega t} \tag{25b}$$

and the normal velocity equations can then be written,

$$v(x,y,t) = [i\omega y_u(x) + u_\infty \frac{\partial y_u}{\partial x}]\, e^{i\omega t} \qquad \text{at } y = 0^+ \tag{26a}$$

$$v(x,y,t) = [i\omega y_L(x) + u_\infty \frac{\partial y_L}{\partial x}]\, e^{i\omega t} \qquad \text{at } y = 0^- \tag{26b}$$

Since the flow is identical between any two adjacent blades,

$$v(x,y,t) = [i\omega y_L(x) + u_\infty \frac{\partial y_L}{\partial x}] e^{i\omega t} \qquad \text{at } y = d^- \qquad (27)$$

In plunge, the lower blade oscillation about $y = 0$ is,

$$y = - h_o e^{i\omega t} \qquad (28)$$

hence,

$$v(x,y,t) = [-i\omega h_o] e^{i\omega t} \qquad \text{at } y = 0^+ \qquad (29)$$

while the upper blade oscillation about $y = d$ is,

$$y = - h_o e^{i(\omega t+\delta)} \qquad (30)$$

where δ is the interblade phase angle. Hence,

$$v(x,y,t) = [\omega \sin \delta\ h_o - i\omega \cos \delta\ h_o] e^{i\omega t} \qquad (31)$$

at $y = d^-$.

In pitch, the lower blade oscillation about $y = 0$ is,

$$y = - \theta_o(x-x_o) e^{i\omega t} \qquad (32)$$

hence,

$$v(x,y,t) = [-u_\infty \theta_o - i\omega\theta_o (x - x_o)] e^{i\omega t} \qquad (33)$$

at $y = 0^+$, while the upper blade oscillation about $y = d$ is,

$$y = -\theta_o (x - B - x_o) e^{i(\omega t+\delta)} \qquad (34)$$

hence,

$$v(x,y,t) = [-\theta_o \{u_\infty \cos \delta - \omega \sin \delta (x-B-x_o)\} - i\theta_o \{u_\infty \sin \delta + \omega \cos \delta (x-B-x_o)\}] e^{i\omega t} \qquad (35)$$

at $y = d^-$.

The pressure distribution on the blades can be written in terms of the local sonic velocity as follows: In terms of perturbation quantities, Eq. (11) can be written,

$$c_\infty^2 + 2c\,c_\infty = \gamma \frac{1}{\rho_\infty} \frac{p_\infty + \Delta p}{(1 + \frac{\Delta\rho}{\rho_\infty})} \tag{36}$$

cancelling the higher order c^2 term. Since $\frac{\Delta\rho}{\rho_\infty}$ is less than one, Eq. (36) can be expanded in the geometric series,

$$\frac{1}{(1+ \frac{\Delta\rho}{\rho_\infty})} = 1 - \frac{\Delta\rho}{\rho_\infty} + \frac{\Delta\rho^2}{\rho_\infty^2} - \dots\dots = 1 - \frac{\Delta\rho}{\rho_\infty} \tag{37}$$

neglecting the higher order terms. Substituting this in Eq. (36) gives,

$$2c\,c_\infty = \gamma \frac{1}{\rho_\infty} (1 - \frac{c_\infty^2}{\gamma} \frac{\Delta\rho}{\Delta p}) \Delta p \tag{38}$$

Since $\Delta\rho$ and Δp are small,

$$\frac{\Delta p}{\Delta\rho} \to \frac{1}{\frac{dp}{d\rho}} = \frac{1}{\bar{c}^2} \tag{39}$$

Substituting this in Eq. (38) gives after some algebraic manipulation,

$$\frac{2}{\gamma-1} c\,c_\infty \left(\frac{1}{1 + \frac{2c}{(\gamma-1)c_\infty}}\right) = \frac{1}{\rho_\infty} \Delta p \tag{40}$$

Since $2c/(\gamma-1)c_\infty$ is small, $\frac{1}{1 + \frac{2c}{(\gamma-1)c_\infty}}$ may be expanded in the geometric series,

$$\frac{1}{1 + \frac{2c}{(\gamma-1)c_\infty}} = 1 - \frac{2c}{(\gamma-1)c_\infty} + \left[\frac{2c}{(\gamma-1)c_\infty}\right]^2 - \dots\dots = 1 - \frac{2c}{(\gamma-1)c_\infty} \tag{41}$$

neglecting higher order terms. Hence, Eq. (40) can be written, neglecting higher order terms,

$$p - p_\infty = \frac{2}{\gamma-1} \rho_\infty c_\infty (\bar{c} - c_\infty) \tag{42}$$

Finally, any non-dimensional quantities so stated in the paper are made such with reference to the blade chord length and the freestream velocity unless otherwise defined.

III. METHOD OF CHARACTERISTICS

In the method of characteristics it is desirable to ascertain the coordinate system across which all possible discontinuities in flow properties may occur. Along this coordinate system the equations of motion of the flow field can then be treated as ordinary differential equations that are solvable by classical or numerical techniques (e.g., finite differences). To obtain the equations of this coordinate system (the characteristic directions) in the (x,y) plane, the equations of motion are written in terms of the arbitrary intersecting coordinates,

$$\xi = \xi\ (x,y) \tag{43a}$$

and

$$\eta = \eta\ (x,y) \tag{43b}$$

If the first derivatives of the dependent variables, $\bar{u}$, v, and p (or $\bar{c}$) with respect to ξ are made indeterminate across lines of η = constant, and the first derivatives of the dependent variables with respect to η are made indeterminate across lines of ξ = constant, then any possible discontinuities in the first derivatives will occur across these lines. These lines are then the characteristics and their equations are obtained in the evaluation of the indeterminacies.

Consider, first, two-dimensional, steady flow. The governing equations of motion are the continuity equation,

$$\frac{\partial(\rho\bar{u})}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{44}$$

the Euler equations,

$$\bar{u}\,\frac{\partial \bar{u}}{\partial x} + v\,\frac{\partial \bar{u}}{\partial y} + \frac{1}{\rho}\,\frac{\partial p}{\partial x} = 0 \tag{45a}$$

$$\bar{u}\,\frac{\partial v}{\partial x} + v\,\frac{\partial v}{\partial y} + \frac{1}{\rho}\,\frac{\partial p}{\partial y} = 0 \tag{45b}$$

and the energy equation,

$$\bar{u}\,\frac{\partial s}{\partial x} + v\,\frac{\partial s}{\partial y} = 0 \tag{46}$$

Along a streamline this equation may be written,

$$\bar{u}\,\frac{\partial p}{\partial x} + v\,\frac{\partial p}{\partial y} = \bar{c}^2[\bar{u}\,\frac{\partial \rho}{\partial x} + v\,\frac{\partial \rho}{\partial y}] \tag{47}$$

Substituting in Eq. (44), the continuity equation becomes,

$$\rho\,[\frac{\partial \bar{u}}{\partial x} + \frac{\partial v}{\partial y}] = -\,\frac{1}{\bar{c}^2}\,[\bar{u}\,\frac{\partial p}{\partial x} + v\,\frac{\partial p}{\partial y}] \tag{48}$$

In terms of the arbitrary coordinates, Eqs. (43), the continuity equation becomes,

$$\rho\xi_x\,\bar{u}_\xi + \rho\xi_y\,v_\xi + \frac{1}{\bar{c}^2}\,[\bar{u}\xi_x + v\xi_y]\,p_\xi$$

$$= -\,\rho\eta_x\,\bar{u}_\eta - \rho\eta_y\,v_\eta - \frac{1}{\bar{c}^2}\,[\bar{u}\,\eta_x + v\,\eta_y]\,p_\eta \tag{49}$$

In like manner, the Euler equations become,

$$\rho[\bar{u}\xi_x + v\xi_y]\,\bar{u}_\xi + \xi_x p_\xi = -\rho[\bar{u}\eta_x + v\eta_y]\,\bar{u}_\eta - p_\eta\eta_x \tag{50a}$$

$$\rho[\bar{u}\xi_x + v\xi_y]\,v_\xi + \xi_y p_\xi = -\rho[\bar{u}\eta_x + v\eta_y]\,v_\eta - p_\eta\eta_y \tag{50b}$$

Eq. (49) and Eqs. (50) form a system of three simultaneous equations in $\bar{u}_\xi$, v_ξ, and p_ξ. Solving for p_ξ by Cramer's rule gives a ratio of two determinants. The denominator is the determinant of the matrix of coefficients of $\bar{u}_\xi$, v_ξ, and p_ξ formed by Eq. (49) and Eqs. (50), while the numerator is

this determinant with the right hand side of the equations substituted for the coefficients of p_ξ.

Since across lines of η = constant, p_ξ is indeterminate, both the numerator and the denominator must equal zero. Setting the denominator equal to zero and expanding the determinant gives,

$$[\bar{u}\xi_x + v\xi_y]\,[(\bar{u}^2 - \bar{c}^2)\,\xi_x^2 + 2\bar{u}v\,\xi_x\xi_y + (v^2 - \bar{c}^2)\xi_y^2] = 0 \tag{51}$$

The solutions to this equation give the equations of all three characteristics in the physical (x,y) plane. These are,

$$\frac{dy}{dx} = \frac{v}{u} = \tan\zeta \tag{52a}$$

$$\frac{dy}{dx} = \tan(\zeta \pm \alpha) \tag{52b}$$

where ζ is the angle the streamline makes with the x-axis and α is the Mach angle. All three characteristic directions are obtained from Eq. (51) because ξ and η are arbitrary and interchangeable: If Eq. (49) and Eqs. (50) were written such that derivatives with respect to η are the unknowns, making p_η indeterminate across lines of ξ = constant would result in solutions identical to Eqs. (52), a priori. Eq. (52a) is merely the equation of a streamline while Eqs. (52b) are the equations of the right and left-running Mach lines. Traditionally, Eqs. (52b) are the equations of ξ = constant and η = constant.

The compatibility relation which relates changes in $\bar{u}$, v, and p along the (ξ,η) characteristics can be obtained by

setting the numerator of p_ξ equal to zero. Expanding the determinant and solving the resultant equation yields after some algebraic manipulation,

$$v \frac{\partial \bar{u}}{\partial \eta} - u \frac{\partial v}{\partial \eta} \mp \frac{1}{\rho} \cot \alpha \frac{\partial p}{\partial \eta} = 0 \tag{53}$$

This derivation is shown in detail in Chapter 8 of Oswatitsch (1956). These relations must be satisfied along the (ξ,η) characteristics.

The physical significance of these characteristics is that they act as boundaries across which any discontinuities will occur. In this flow field where streamlines and Mach lines are the characteristics, a discontinuity is readily apparent across the Mach line emanating from the leading-edge of a slender airfoil or across a slip plane (streamline) emanating from a Mach stem. Characteristics can further be described as information carriers in that along them all disturbances propagate and all flow quantities can be determined given their values in the domain of dependence. It should also be noted that the characteristic directions are completely independent of the type of coordinate system used to describe the flow. They are determined solely by the physical nature of the flow field as expressed in the equations of motion.

The problem of two-dimensional, unsteady flow over a flat plate as shown by Teipel (1962) can be considered in the same manner as before. The governing equations of motion [Eq. (2), Eq. (3), and Eq. (4)] are written in an arbitrary coordinate

system across which variation of flow properties are indeterminate. The evaluation of these indeterminacies will again give the characteristic directions of these coordinates.

The arbitrary coordinates are given in Eqs. (48). Eq.(2), Eq. (3), and Eq. (4) can be rewritten in a form more suitable for transformation by using the local sonic velocity as an unknown. Assuming that air is a perfect gas, Eq. (11) can be written,

$$\bar{c}^2 = \gamma \frac{\mathcal{R}}{\mathcal{M}} T \tag{54}$$

where $\mathcal{R}$ is the universal gas constant and $\mathcal{M}$ is the molecular weight. Taking the total differential and dividing by $\bar{c}^2$ gives,

$$2 \frac{d\bar{c}}{\bar{c}} = \gamma \frac{\mathcal{R}}{\mathcal{M}} \frac{dT}{\bar{c}^2} \tag{55}$$

Substituting Eq. (54) into this gives,

$$2 \frac{d\bar{c}}{\bar{c}} = \frac{dT}{T} \tag{56}$$

In like manner, the total differential of Eq. (11) is,

$$2\bar{c}\, d\bar{c} = \gamma \frac{p}{\rho} \cdot \frac{dp}{p} - \gamma \frac{p}{\rho} \cdot \frac{d\rho}{\rho} \tag{57}$$

and from Eq. (11) again,

$$2 \frac{d\bar{c}}{\bar{c}} = \frac{dp}{p} - \frac{d\rho}{\rho} \tag{58}$$

From the First Law of Thermodynamics and the definition of entropy,

$$T ds = du - \frac{p}{\rho^2} d\rho \tag{59}$$

Since $du = c_v \cdot dt$, this may be written as, (60)

$$Tds = c_v dT - RT \frac{d\rho}{\rho} \tag{61}$$

where R is the gas constant $(c_p - c_v)$ and c_p and c_v are the specific heats of the gas at constant pressure and volume, respectively. Dividing Eq. (61) by Tc_v, substituting from Eq. (56) and Eq. (59), and then taking the substantial derivative of $\frac{1}{c_v}S$ with respect to time gives,

$$\frac{1}{c_v} \frac{Ds}{Dt} = 2\gamma \frac{1}{\bar{c}} \frac{D\bar{c}}{Dt} - (\gamma - 1) \frac{1}{p} \frac{Dp}{Dt} = 0 \tag{62}$$

since entropy is held constant. This can be written substituting from Eq. (11),

$$\frac{Dp}{Dt} = \frac{2}{\gamma - 1} \rho \bar{c} \frac{D\bar{c}}{Dt} \tag{63}$$

Introducing the small perturbation assumption and substituting from Eq. (7) and Eq. (63), Eq. (2) can be written, cancelling higher order terms,

$$\frac{2}{\gamma - 1} \frac{\partial \bar{c}}{\partial t} + \frac{2}{\gamma - 1} u_\infty \frac{\partial \bar{c}}{\partial x} + c_\infty \frac{\partial \bar{u}}{\partial x} + c_\infty \frac{\partial v}{\partial y} = 0 \tag{64}$$

This is the form of the continuity equation that will be used in the coordinate transformation. Again dividing Eq. (61) by Tc_v, substituting from Eq. (56) and Eq. (58), and then taking the partial derivative with respect to x (holding entropy constant) gives,

$$\frac{1}{\rho} \frac{\partial p}{\partial x} = \frac{2}{\gamma - 1} \bar{c} \frac{\partial \bar{c}}{\partial x} \tag{65}$$

Substituting in the first Euler equation, Eq. (3) becomes, cancelling higher order terms,

$$\frac{\partial \bar{u}}{\partial t} + u_\infty \frac{\partial \bar{u}}{\partial x} + \frac{2}{\gamma - 1} c_\infty \frac{\partial \bar{c}}{\partial x} = 0 \tag{66}$$

In like manner, the second Euler equation, Eq. (4), becomes,

$$\frac{\partial v}{\partial t} + u_\infty \frac{\partial v}{\partial x} + \frac{2}{\gamma - 1} c_\infty \frac{\partial \bar{c}}{\partial y} = 0 \tag{67}$$

The continuity equation and the Euler equations thus form a system of three simultaneous equations in $\bar{u}$, v, and $\bar{c}$. With assumption of simple harmonic motion, Teipel (1962) introduced his amplitude functions,

$$U(x,y)e^{i\omega t} = \frac{\bar{u} - u_\infty}{u_\infty} \tag{68}$$

$$V(x,y)e^{i\omega t} = \frac{1}{\sqrt{M^2 - 1}} \frac{v}{u_\infty} \tag{69}$$

$$C(x,y)e^{i\omega t} = \frac{2}{\gamma - 1} \frac{1}{M^2} \frac{\bar{c} - c_\infty}{c_\infty} \tag{70}$$

where U, V, and C are complex.

Substituting in Eq. (64), Eq. (66), and Eq. (67) gives,

$$\frac{\partial U}{\partial x} + \sqrt{M^2 - 1} \frac{\partial V}{\partial y} + M^2 \frac{\partial C}{\partial x} + ikM^2 C = 0 \tag{71}$$

$$\frac{\partial U}{\partial x} + \frac{\partial C}{\partial x} + ikU = 0 \tag{72}$$

$$\frac{\partial V}{\partial x} + \frac{1}{\sqrt{M^2 - 1}} \frac{\partial C}{\partial y} + ikV = 0 \tag{73}$$

where k is the reduced frequency.

Transforming this system into the new coordinate system, $\xi(x,y)$ and $\eta(x,y)$, and considering now a system of three simultaneous equations in unknowns, U_ξ, V_ξ, and C_ξ, the equations may be written, (74)

$$\xi_x U_\xi + \cot\alpha\, \xi_y V_\xi + M^2 \xi_x C_\xi = -U_\eta - \cot\alpha\, V_\eta \eta_y - M^2 C_\eta \eta_x - ikM^2 C$$

$$\xi_x U_\xi + \xi_x C_\xi = -U_\eta \eta_x - C_\eta \eta_x - ikU \tag{75}$$

$$\xi_x V_\xi + \tan\alpha\, \xi_y C_\xi = -V_\eta \eta_x - \tan\alpha\, C_\eta \eta_y - ikV \qquad (76c)$$

As before U_ξ, V_ξ, and C_ξ are made indeterminate across lines of η = constant. This forces any discontinuities in U, V, and C to lie across lines of η = constant. One condition of indeterminacy is that the determinant of the matrix of coefficients of U_ξ, V_ξ, and C_ξ in the above equations be zero. If this determinant is expanded and then set equal to zero, the resulting equation is,

$$-\xi_x^3 - \xi_x [\xi_y^2 - M^2\xi_x^2] = 0 \qquad (77)$$

As before the solutions to this equation give the three characteristic directions in the physical (x,y) plane:

$$\frac{dy}{dx} = 0 \qquad (78a)$$

$$\frac{dy}{dx} = \pm \tan\alpha \qquad (78b)$$

These coincide with the results of the steady flow analysis when it is remembered that the small perturbation assumption forces the streamlines to be parallel with the x-axis ($\zeta = 0$).

In order to find the compatibility relations of the first derivatives of U, V, and C along the characteristics, the equations of motion are written along lines of ξ = constant, η = constant, and along streamlines, that is, for

$$str(x,y) = \text{constant}, \qquad \frac{\partial y}{\partial x} = 0 \qquad (79a)$$

$$\xi(x,y) = \text{constant}, \qquad \frac{\partial y}{\partial x} = \frac{1}{\sqrt{M^2-1}} \qquad (79b)$$

$$\eta(x,y) = \text{constant}, \quad \frac{\partial y}{\partial x} = -\frac{1}{\sqrt{M^2-1}} \qquad (79c)$$

If an arbitrary function of x and y in the equations is denoted by f then, $\frac{\partial f}{\partial x}$ holding str constant becomes,

$$\left(\frac{\partial f}{\partial x}\right)_{str} = \frac{\partial f}{\partial x} \qquad (80)$$

Likewise, $\frac{\partial f}{\partial x}$ holding ξ constant becomes,

$$\left(\frac{\partial f}{\partial x}\right)_{\xi} = \frac{\partial f}{\partial x} + \frac{1}{\sqrt{M^2-1}} \frac{\partial f}{\partial y} \qquad (81)$$

and, $\frac{\partial f}{\partial x}$ holding η constant becomes,

$$\left(\frac{\partial f}{\partial x}\right)_{\eta} = \frac{\partial f}{\partial x} - \frac{1}{\sqrt{M^2-1}} \frac{\partial f}{\partial y} \qquad (82)$$

Combining these equations and rewriting,

$$\frac{\partial f}{\partial x} = \left(\frac{\partial f}{\partial x}\right)_{str} \qquad (83a)$$

$$\frac{\partial f}{\partial x} = \frac{1}{2}\left[\left(\frac{\partial f}{\partial x}\right)_{\xi} + \left(\frac{\partial f}{\partial x}\right)_{\eta}\right] \qquad (83b)$$

$$\frac{\partial f}{\partial y} = \frac{1}{2}\sqrt{M^2-1}\left[\left(\frac{\partial f}{\partial x}\right)_{\xi} - \left(\frac{\partial f}{\partial x}\right)_{\eta}\right] \qquad (83c)$$

Substituting these relations in the equations of motion gives,

$$\frac{1}{2}\left[\left(\frac{\partial U}{\partial x}\right)_{\xi} + \left(\frac{\partial U}{\partial x}\right)_{\eta}\right] + \frac{1}{2}(M^2-1)\left[\left(\frac{\partial V}{\partial x}\right)_{\xi} - \left(\frac{\partial V}{\partial x}\right)_{\eta}\right]$$

$$+ \frac{1}{2} M^2\left[\left(\frac{\partial C}{\partial x}\right)_{\xi} + \left(\frac{\partial C}{\partial x}\right)_{\eta}\right] + ikM^2C = 0 \qquad (84a)$$

$$\frac{1}{2}\left[\left(\frac{\partial U}{\partial x}\right)_{\xi} + \left(\frac{\partial U}{\partial x}\right)_{\eta}\right] + \frac{1}{2}\left[\left(\frac{\partial C}{\partial x}\right)_{\xi} + \left(\frac{\partial C}{\partial x}\right)_{\eta}\right] + ikU = 0 \qquad (84b)$$

$$\left[\left(\frac{\partial V}{\partial x}\right)_{\xi} + \left(\frac{\partial V}{\partial x}\right)_{\eta}\right] + \frac{1}{2}\left[\left(\frac{\partial C}{\partial x}\right)_{\xi} - \left(\frac{\partial C}{\partial x}\right)_{\eta}\right] + ikV = 0 \qquad (84c)$$

Subtracting Eq. (84b) from Eq. (84a) gives an equation that is first added to Eq. (84c) and then subtracted from Eq. (84c). The result is two compatibility relations that must be

satisfied along lines of constant ξ and lines of constant η. These are,

$$\left(\frac{\partial V}{\partial x}\right)_\xi + \left(\frac{\partial C}{\partial x}\right)_\xi + ik\left[V + \frac{1}{M^2-1}(M^2C - U)\right] = 0 \qquad (85a)$$

$$\left(\frac{\partial V}{\partial x}\right)_\eta - \left(\frac{\partial C}{\partial x}\right)_\eta + ik\left[V - \frac{1}{M^2-1}(M^2C - U)\right] = 0 \qquad (85b)$$

The third compatibility relation is obtained by considering Eq. (84b) along a streamline,

$$\left(\frac{\partial U}{\partial x}\right)_{str} + \left(\frac{\partial C}{\partial x}\right)_{str} + ikU = 0 \qquad (86)$$

If irrotationality is assumed, the equation of irrotationality,

$$\frac{\partial \bar{u}}{\partial y} - \frac{\partial v}{\partial x} = 0 \qquad (87)$$

may be substituted for the second Euler equation in the system. Again with the assumption of simple harmonic motion, it may be written in terms of Teipel's amplitude functions,

$$\frac{\partial U}{\partial y} - \sqrt{M^2 - 1}\,\frac{\partial V}{\partial x} = 0 \qquad (88)$$

The non-dimensional system of equations can then be written as,

$$\frac{\partial U}{\partial x} + \sqrt{M^2 - 1}\,\frac{\partial V}{\partial y} + M^2\frac{\partial C}{\partial x} + ikM^2C = 0 \qquad (89a)$$

$$\frac{\partial U}{\partial x} + \frac{\partial C}{\partial x} + ikU = 0 \qquad (89b)$$

$$\frac{\partial U}{\partial y} - \sqrt{M^2 - 1}\,\frac{\partial V}{\partial x} = 0 \qquad (89c)$$

By transforming Eqs. (89) to the arbitrary coordinate system, ξ and η, as before, the system can be considered one of simultaneous equations of U_ξ, V_ξ, and C_ξ. By making any of these

derivatives indeterminate the results of setting the determinant of the matrix of their coefficients equal to zero are Eqs. (78), the characteristic directions.

By writing the equations of motion, as before, along lines of ξ = constant and η = constant and performing the same algebraic manipulations, the two compatibility relations can be determined:

$$\left(\frac{\partial U}{\partial x}\right)_\xi - \left(\frac{\partial V}{\partial x}\right)_\xi + ik\frac{M^2}{M^2-1}(U - C) = 0 \tag{90a}$$

$$\left(\frac{\partial U}{\partial x}\right)_\eta + \left(\frac{\partial V}{\partial x}\right)_\eta + ik\frac{M^2}{M^2-1}(U - C) = 0 \tag{90b}$$

The third compatibility relation is, as before,

$$\left(\frac{\partial U}{\partial x}\right)_{str} + \left(\frac{\partial C}{\partial x}\right)_{str} + ikU = 0 \tag{86}$$

The equations of motion are now reduced to a system containing only derivatives with respect to x. It is, thus, possible to solve these equations using finite differences in the following manner: Separating the real from the imaginary components, the equations can be written,

$$\left(\frac{\partial U_R}{\partial x}\right)_{str} + \left(\frac{\partial C_R}{\partial x}\right)_{str} - kU_I = 0 \tag{91a}$$

$$\left(\frac{\partial U_I}{\partial x}\right)_{str} + \left(\frac{\partial C_I}{\partial x}\right)_{str} + kU_R = 0 \tag{91b}$$

$$\left(\frac{\partial U_R}{\partial x}\right)_\xi - \left(\frac{\partial V_R}{\partial x}\right)_\xi - k\frac{M^2}{M^2-1}(U_I - C_I) = 0 \tag{91c}$$

$$\left(\frac{\partial U_I}{\partial x}\right)_\xi - \left(\frac{\partial V_I}{\partial x}\right)_\xi + k\frac{M^2}{M^2-1}(U_R - C_R) = 0 \tag{91d}$$

$$\left(\frac{\partial U_R}{\partial x}\right)_\eta + \left(\frac{\partial V_R}{\partial x}\right)_\eta - k\frac{M^2}{M^2-1}(U_I - C_I) = 0 \tag{91e}$$

$$\left(\frac{\partial U_I}{\partial x}\right)_\eta + \left(\frac{\partial V_I}{\partial x}\right)_\eta + k \frac{M^2}{M^2-1} (U_R - C_R) = 0 \tag{91f}$$

To write these equations in finite difference form, a computational molecule as shown in Figure 2 will be used. With the velocity values at P_{11}, P_{12}, and P_{21} known, all the values at P_{22} can be calculated. The distance Δx is arbitrary.

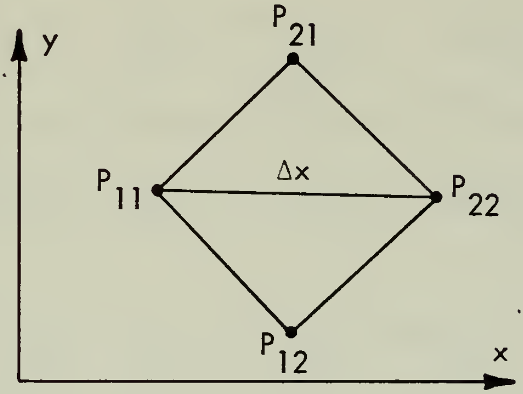


Figure 2

If F is an arbitrary flow quantity, its partial derivatives can thus be written,

$$\left(\frac{\partial F}{\partial x}\right)_{str} = \frac{F_{22} - F_{11}}{\Delta x} \tag{92a}$$

$$\left(\frac{\partial F}{\partial x}\right)_\xi = \frac{F_{22} - F_{12}}{\frac{1}{2}\Delta x} \tag{92b}$$

$$\left(\frac{\partial F}{\partial x}\right)_\eta = \frac{F_{22} - F_{21}}{\frac{1}{2}\Delta x} \tag{92c}$$

The values of the flow quantities in the equations are averages, such that,

$$(F)_{str} = \frac{1}{2} (F_{11} + F_{22}) \tag{93a}$$

$$(F)_{\xi} = \frac{1}{2}(F_{12} + F_{22}) \tag{93b}$$

$$(F)_{\eta} = \frac{1}{2}(F_{21} + F_{22}) \tag{93c}$$

With these relations substituted into the equations, the system becomes, after some algebraic manipulation,

$$U_{22R} + C_{22R} - A_I U_{22I} = K_{12R} \tag{94a}$$

$$U_{22I} + C_{22I} + A_I U_{22R} = K_{12I} \tag{94b}$$

$$U_{22R} - V_{22R} - B_I U_{22I} + B_I C_{22I} = K_{34R} \tag{94c}$$

$$U_{22I} - V_{22I} + B_I U_{22R} - B_I C_{22R} = K_{34I} \tag{94d}$$

$$U_{22R} + V_{22R} - B_I U_{22I} + B_I C_{22I} = K_{56R} \tag{94e}$$

$$U_{22I} + V_{22I} + B_I U_{22R} - B_I C_{22R} = K_{56I} \tag{94f}$$

where:

$$K_{12R} = U_{11R} + C_{11R} + A_I U_{11I} \tag{95a}$$

$$K_{12I} = U_{11I} + C_{11I} - A_I U_{11R} \tag{95b}$$

$$K_{34R} = U_{12R} - V_{12R} + B_I(U_{12I} - C_{12I}) \tag{95c}$$

$$K_{34I} = U_{12I} - V_{12I} - B_I(U_{12R} - C_{12R}) \tag{95d}$$

$$K_{56R} = U_{21R} + V_{21R} + B_I(U_{21I} - C_{21I}) \tag{95e}$$

$$K_{56I} = U_{21I} + V_{21I} - B_I(U_{21R} - C_{21R}) \tag{95f}$$

and,

$$A_I = \frac{1}{2} k \, \Delta x \tag{96a}$$

$$B_I = \frac{1}{4} k \frac{M^2}{M^2 - 1} \Delta x \tag{96b}$$

As shown in Teipel (1962), solving these equations gives,

$$U_{22R}=\frac{(1-A_IB_I)[\frac{1}{2}(K_{34R}+K_{56R})-B_IK_{12I}]+2B_I[\frac{1}{2}(K_{34I}+K_{56I})+B_IK_{12R}]}{(1-A_IB_I)^2+(2B_I)^2} \qquad (97a)$$

$$U_{22I}=\frac{(1-A_IB_I)[\frac{1}{2}(K_{34I}+K_{56I})+B_IK_{12R}]-2B_I[\frac{1}{2}(K_{34R}+K_{56R})-B_IK_{12I}]}{(1-A_IB_I)^2+(2B_I)^2} \qquad (97b)$$

$$V_{22R} = \frac{1}{2}(K_{56R}-K_{34R}) \qquad (97c)$$

$$V_{22I} = \frac{1}{2}(K_{56I}-K_{34I}) \qquad (97d)$$

$$C_{22R} = K_{12R}-U_{22R}+A_IU_{22I} \qquad (97e)$$

$$C_{22I} = K_{12I}-U_{22I}-A_IU_{22R} \qquad (97f)$$

These are the finite difference equations for a point in the general flow field.

At a point on the top of a cascade blade or solid boundary, the normal velocity V_{22} is prescribed by the movement of the blade. This velocity was given in the Problem Formulation. Here the computational molecule is shown in Figure 3.

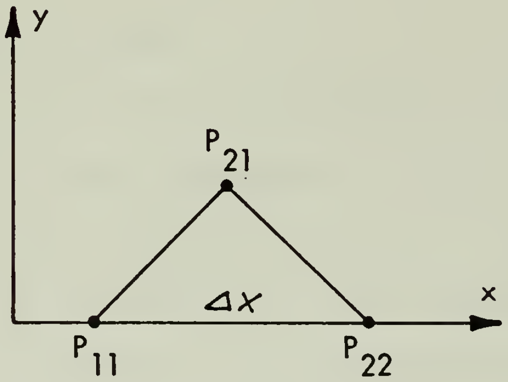


Figure 3

The applicable equations are,

$$U_{22R} + C_{22R} - A_I U_{22I} = K_{12R} \tag{98a}$$

$$U_{22I} + C_{22I} + A_I U_{22R} = K_{12I} \tag{98b}$$

$$U_{22R} - B_I U_{22I} + B_I C_{22I} = K_{56R} - K_{34R} \tag{98c}$$

$$U_{22I} + B_I U_{22R} - B_I C_{22R} = K_{56I} - K_{34I} \tag{98d}$$

where K_{12} and K_{56} are as before, but,

$$K_{34R} = V_{22R} \tag{99a}$$

$$K_{34I} = V_{22I} \tag{99b}$$

as given by the boundary conditions. Solving for these equations gives,

$$U_{22R} = \frac{(1-A_I B_I)[K_{56R}-K_{34R}-B_I K_{12I}] + 2B_I[K_{56I}-K_{34I}+B_I K_{12R}]}{(1-A_I B_I)^2 + (2B_I)^2} \tag{100a}$$

$$U_{22I} = \frac{(1-A_I B_I)[K_{56I}-K_{34I}+B_I K_{12R}] - 2B_I[K_{56R}-K_{34R}-B_I K_{12I}]}{(1-A_I B_I)^2 + (2B_I)^2} \tag{100b}$$

$$C_{22R} = K_{12R} - U_{22R} + A_I U_{22I} \tag{100c}$$

$$C_{22I} = K_{12I} - U_{22I} - A_I U_{22R} \tag{100d}$$

V_{22R} and V_{22I} are known from the boundary conditions. These are the finite difference equations for the flow velocities at the top of a solid boundary or cascade blade.

Likewise, at the bottom of a solid boundary or cascade blade, the normal velocity, V_{22} is prescribed by the movement of the blade, and was given in the Problem Formulation. The

computational molecule, different from that at the top of a blade, is shown in Figure 4.

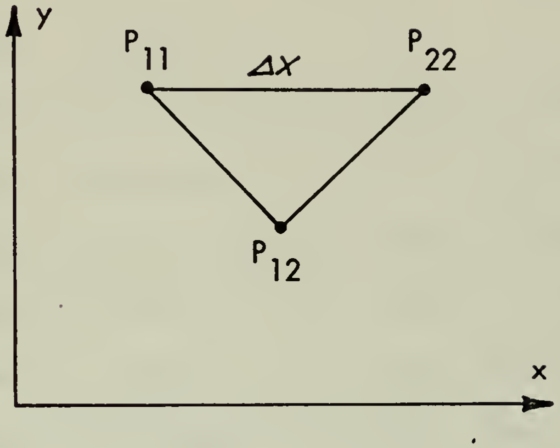


Figure 4

Here, the applicable equations are,

$$U_{22R} + C_{22R} - A_I U_{22I} = K_{12R} \quad (101a)$$

$$U_{22I} + C_{22I} - A_I U_{22R} = K_{12I} \quad (101b)$$

$$U_{22R} - B_I U_{22I} + B_I C_{22I} = K_{56R} + K_{34R} \quad (101c)$$

$$U_{22I} + B_I U_{22R} - B_I C_{22R} = K_{56I} + K_{34I} \quad (101d)$$

where K_{12} and K_{34} are as originally stated, but,

$$K_{56R} = V_{22R} \quad (102a)$$

$$K_{56I} = V_{22I} \quad (102b)$$

as given by the boundary conditions. Thus, the finite difference equations for the flow velocities at the bottom of a solid boundary or cascade blade are,

$$U_{22R} = \frac{(1-A_I B_I)[K_{56R}+K_{34R}-B_I K_{12I}]+2B_I[K_{56I}+K_{34I}+B_I K_{12R}]}{(1-A_I B_I)^2 + (2B_I)^2} \quad (103a)$$

$$U_{22I} = \frac{(1-A_I B_I)[K_{56I}+K_{34I}+B_I K_{12R}]-2B_I[K_{56R}+K_{34R}-B_I K_{12I}]}{(1-A_I B_I)^2 + (2B_I)^2} \qquad (103b)$$

$$C_{22R} = K_{12R} - U_{22R} + A_I U_{22I} \qquad (103c)$$

$$C_{22I} = K_{12I} - U_{22I} - A_I U_{22R} \qquad (103d)$$

V_{22R} and V_{22I} are given from the boundary condition.

To obtain the conditions on the initial left-running Mach line, assume that P_{11} and P_{21} as shown in Figure 2 are just in the freestream and then let Δx shrink to zero. Since $K_{12} = K_{56} = 0$ and A_I and B_I go to zero as Δx goes to zero, the initial finite difference equations for ξ = constant are,

$$U_{22} + C_{22} = 0 \qquad (104a)$$

$$U_{22} - V_{22} + V_{12} = 0 \qquad (104b)$$

$$U_{22} + V_{22} = 0 \qquad (104c)$$

Thus,

$$U_{22} = - V_{22} = - C_{22} \qquad (105)$$

Eq. (90a) can then be written,

$$\left(\frac{\partial U}{\partial x}\right)_{\xi} = - ikU \qquad (106)$$

Integrating gives,

$$U = U\Big|_{x=0} \exp[-ik \frac{M^2}{M^2-1} x] \qquad (107)$$

Thus,

$$U_{22R} = - V_{22R}(0)\cos\left(k \frac{M^2}{M^2-1} x\right) - V_{22I}(0)\sin\left(k \frac{M^2}{M^2-1} x\right) \qquad (108a)$$

$$U_{22I} = -V_{22I}(0)\cos\left(k \frac{M^2}{M^2-1} x\right) + V_{22R}(0)\sin\left(k \frac{M^2}{M^2-1} x\right) \qquad (108b)$$

$$V_{22R} = - U_{22R} \tag{108c}$$

$$V_{22I} = - U_{22I} \tag{108d}$$

$$C_{22R} = - U_{22R} \tag{108e}$$

$$C_{22I} = - U_{22I} \tag{108f}$$

For the initial right-running Mach line, P_{11} and P_{12} of Figure 2 are just in the freestream and Δx is brought to zero. Since $K_{12} = K_{34} = 0$, and $A_I = B_I = 0$,

$$U_{22} + C_{22} = 0 \tag{109a}$$

$$U_{22} - V_{22} = 0 \tag{109b}$$

$$U_{22} - U_{21} + V_{22} = 0 \tag{109c}$$

Thus,

$$U_{22} = V_{22} = - C_{22} \tag{110}$$

Eq. (90b) can then be written,

$$\left(\frac{\partial U}{\partial x}\right)_\eta = - ikU \tag{111}$$

Integrating gives,

$$U = V(0)\ e^{[-ik \frac{M^2}{M^2 - 1} x]} \tag{112}$$

Thus,

$$U_{22R} = V_{22R}(0) \cos(k \frac{M^2}{M^2 - 1} x) + V_{22I}(0) \sin(k \frac{M^2}{M^2 - 1} x) \tag{113a}$$

$$U_{22I} = V_{22I}(0) \cos(k \frac{M^2}{M^2 - 1} x) - V_{22R}(0) \sin(k \frac{M^2}{M^2 - 1} x) \tag{113b}$$

$$V_{22R} = U_{22R} \tag{113c}$$

$$C_{22R} = -U_{22R} \tag{113d}$$

$$V_{22I} = U_{22I} \tag{113e}$$

$$C_{22I} = -U_{22I} \tag{113f}$$

The pressure distribution along the surfaces of the blades can be determined from Eq. (42). Dividing both sides by p_∞ and using Teipel's amplitude function for C, the equation can be written,

$$P(x,y) = \gamma M^2 C(x,y) \Big|_{x=0^-}^{x=0^+} \tag{114}$$

where

$$P(x,y)e^{i\omega t} = \frac{p - p_\infty}{p_\infty}$$

Thus the non-dimensional pressure distribution along the top surface and the bottom surface of the cascade blade can be computed in a point by point calculation of U, V, and C along lines of ξ = constant and η = constant.

IV. ELEMENTARY THEORY FOR SLOWLY OSCILLATING CASCADES

Sauer's (1950) solution for an airfoil oscillating at low frequencies in an unbounded supersonic flow can be applied to the flat plate cascade to form a theory suitable for comparison with the method of characteristics. As shown in Garrick (1957), the equations of motion of supersonic flow over a flat plate oscillating at small amplitude can be written, assuming irrotational flow, as

$$\cot^2\alpha\Phi_{xx} - \Phi_{yy} + 2\frac{1}{c_\infty}M\Phi_{xt} - \frac{1}{c_\infty^2}\Phi_{tt} = 0 \qquad (115)$$

With the assumption of simple harmonic motion,

$$\Phi(x,y,t) = \phi(x,y,k)e^{ikt} \qquad (116)$$

and the potential equation becomes,

$$\cot^2\alpha\phi_{xx} - \phi_{yy} + 2ikM^2\phi_x - k^2M^2\phi = 0 \qquad (117)$$

where ϕ is complex.

If the frequency of oscillation is sufficiently low, the perturbation potential amplitude may be expanded in a Taylor series, such that

$$\phi(x,y,k) = \chi(x,y) + k\,\psi(x,y) \qquad (118)$$

neglecting higher order terms. Eq. (117) then splits into two simultaneous equations since k cannot appear in any relation, it being independent of k. These equations are:

$$\cot^2\alpha\,\chi_{xx} - \chi_{yy} = 0 \qquad (119a)$$

$$\cot^2\alpha\,\psi_{xx} - \psi_{yy} = -2iM^2\chi_x \qquad (119b)$$

Sauer (1950) showed that a general solution to these two equations is,

$$\chi = g(Z) \tag{120}$$

$$\psi = h(Z) - i \frac{M}{\cos \alpha} yg(Z) \tag{121}$$

$$Z = x - y \cot \alpha \tag{122}$$

$$\psi = h(Z) + i \frac{M}{\cos \alpha} yg(Z) \tag{123}$$

$$Z = x + y \cot \alpha \tag{124}$$

where g and h are arbitrary functions of position and are equal to zero for $z \leq 0$ (x,y, z and d are non-dimensional).

To apply this general solution to the cascade with supersonic leading-edge locus, the flow field is divided into separate zones as shown in Figure 5. The number of these zones is dependent on the size of A. For $A > 1$, there is one zone along each blade; for $0.5 < A \leq 1$, there are two; for $0.33 < A \leq 0.5$, there are three; for $0.25 < A \leq 0.33$, there are four; etc.

If oscillation in pitch only is analysed, the boundary conditions on the blade surfaces can be written, in non-dimensional form, from Eq. (33) and Eq. (35),

$$\phi_y = -1 - ik(x - x_o) \qquad \text{at } y = 0 \tag{125}$$

$$\phi_y = e^{i\delta}[-1 - ik(x - x_o)] \qquad \text{at } y = d \tag{126}$$

where θ_o is one.

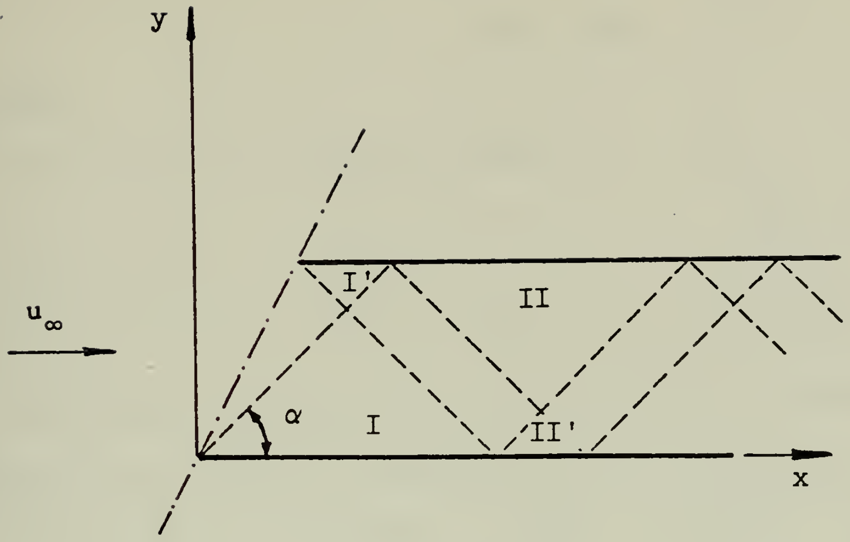


Figure 5

In Zone I there is no interference from the upper blade, hence, Sauer's solution for the single airfoil applies. The boundary condition, however, must be satisfied at $y = 0$, hence,

$$\chi_y = -\cot\alpha\; g_o'(Z) = -1 \tag{127a}$$

$$\psi_y = -\cot\alpha\; h_o'(Z) - i\frac{M}{\cos\alpha}\, g_o(Z) = -i(x - x_o) \tag{127b}$$

This gives,

$$g_o(Z) = Z\tan\alpha \tag{128a}$$

$$h_o(Z) = -i\tan\alpha Z\left(x_o + \frac{Z}{2}\tan^2\alpha\right) \tag{128b}$$

where,

$$\phi = g_o(Z) + k\left\{h_o(Z) - i\frac{M}{\cos\alpha}\, y\, g_o(Z)\right\} \tag{129}$$

In terms of Teipel's amplitude functions the flow properties can be related to those obtained in Method of Characteristics by,

$$U = \phi_x \tag{130a}$$

$$V = \tan\alpha\ \phi_y \tag{130b}$$

$$C = -[\phi_x + ik\phi] \tag{130c}$$

In Zone I,

$$U_I = \tan\alpha\ \{1 - ik[x_o + x\tan^2\alpha + y\cot\alpha]\} \tag{131a}$$

$$V_I = \tan\alpha\ \{-1 + ik[x_o - x + y(\cot\alpha + \frac{M}{\cos\alpha})]\} \tag{131b}$$

$$C_I = \tan\alpha\ \{-1 + ik[x_o + x(\tan^2\alpha - 1) + 2y\cot\alpha]\} \tag{131c}$$

In Zone I' the perturbation potential can also be written directly from Sauer's single airfoil solution,

$$\phi = g_o(\bar{Z}) + k\ \{h_o(\bar{Z}) + i\frac{M}{\cos\alpha}\ y\ g_o(\bar{Z})\} \tag{132}$$

where,

$$\bar{Z} = x - A - B + y\cot\alpha \tag{133}$$

This ensures that $\bar{Z} \leq 0$ upstream of the leading-edge of the upper blade.

The boundary condition at $y = d$ gives,

$$\chi_y = \cot\alpha\ g_o'(\bar{Z}) = -e^{i\delta} \tag{134a}$$

$$\psi_y = \cot\alpha\ h_o'(\bar{Z}) + i\frac{M}{\cos\alpha}\ g_o(\bar{Z}) + i\frac{Md}{\cos\alpha}\cot\alpha g_o'(\bar{Z}) \tag{134b}$$

hence, $\qquad = -i(x - x_o - B)\ e^{i\delta}$

$$g_o(\bar{Z}) = -\bar{Z}\tan\alpha\ e^{i\delta} \tag{135a}$$

$$h_o(\bar{Z}) = i\tan\alpha\ \bar{Z}e^{i\delta}\{x_o + \frac{\bar{Z}}{2}\tan^2\alpha + \frac{Md}{\cos\alpha}\} \tag{135b}$$

Then,

$$U_{I'} = -\tan\alpha\ e^{i\delta}\{1 - ik[x_o + (x-B)\tan^2\alpha + A - y\cot\alpha]\} \tag{136a}$$

$$V_{I'} = -\tan\alpha\ e^{i\delta}\{-1 + ik[x_o - (x-B) + (d-y)(\cot\alpha + \frac{M}{\cos\alpha})]\} \tag{136b}$$

$$C_{I'} = -\tan\alpha\ e^{i\delta}\{-1 + ik[x_o + (x-B)(\tan^2\alpha - 1) + 2A - 2y\cot\alpha]\} \tag{136c}$$

For $0.5 \leq A \leq 1.0$, Zones II and II' will be present in addition to Zones I and I'. In Zone II the perturbation potential is due to the initial waves from Zones I and I' plus the reflected waves at Zone II, hence,

$$\phi = g_o(Z) + g_o(\bar{Z}) + g_1(Z_1) + k\{h_o(Z) + h_o(\bar{Z}) + h_1(Z_1) - i\,\frac{M}{\cos\alpha}\,y[g_o(Z) - g_o(\bar{Z}) - g_1(Z_1)]\} \qquad (137)$$

where Z and $\bar{Z}$ are defined in Eq. (122) and Eq. (133), and,

$$Z_1 = x - 2A + y\cot\alpha \qquad (138)$$

The boundary condition at $y = d$ gives,

$$\chi_y = -\cot\alpha\, g_o'(Z) + \cot\alpha\, g_o'(\bar{Z}) + \cot\alpha\, g_1'(Z_1) = -e^{i\delta} \qquad (139a)$$

$$\begin{aligned}\psi_y = &-\cot\alpha\, h_o'(Z) + \cot\alpha\, h_o'(\bar{Z}) + \cot\alpha\, h_1'(Z_1)\\ &- i\,\frac{M}{\cos\alpha}\,[g_o(Z) - g_o(\bar{Z}) - g_1(Z_1)]\\ &+ i\,\frac{Md}{\cos\alpha}\cot\alpha\,[g_o'(Z) + g_o'(\bar{Z}) + g_1'(Z_1)]\\ &= -i(x - x_o - B)\,e^{i\delta}\end{aligned} \qquad (139b)$$

hence, after some algebraic manipulation

$$g_1(Z_1) = Z_1\tan\alpha \qquad (140a)$$

$$h_1(Z_1) = -i\tan\alpha\, Z_1\{x_o + \frac{Z_1}{2}\tan^2\alpha + 2\,\frac{Md}{\cos\alpha}\} \qquad (140b)$$

Then,

$$\begin{aligned}U_{II} = \tan\alpha\{&2 - e^{i\delta} - ik[(2 - e^{i\delta})(x_o + x\tan^2\alpha + A)\\ &+ e^{i\delta}(y\cot\alpha + B\tan^2\alpha)]\}\end{aligned} \qquad (141a)$$

$$\begin{aligned}V_{II} = \tan\alpha\{&-e^{i\delta} + ik[e^{i\delta}(x_o - x + (d-y)(\cot\alpha + \frac{M}{\cos\alpha}))\\ &-2(d-y)(\cot\alpha + \frac{M}{\cos\alpha})]\}\end{aligned} \qquad (141b)$$

$$\begin{aligned}C_{II} = -\tan\alpha\{&2 - e^{i\delta} + ik[e^{i\delta}(x_o + (x-B)(\tan^2\alpha - 1)\\ &+ 2A - 2y\cot\alpha) - 2(x_o + x(\tan^2\alpha - 1) + 2A)]\}\end{aligned} \qquad (141c)$$

In Zone II' the perturbation potential is due to the initial waves from Zones I and I' plus the reflected waves at Zones II and II', hence,

$$\phi = g_o(Z) + g_o(\bar{Z}) + g_1(\bar{Z}_1) + k\{h_o(Z) + h_o(\bar{Z})$$
$$+ h_1(\bar{Z}_1) - i \frac{M}{\cos\alpha} y[g_o(Z) - g_o(\bar{Z}) + g_1(\bar{Z}_1)]\} \qquad (142)$$

where Z and $\bar{Z}$ are defined in Eq. (122) and Eq. (133), and,

$$\bar{Z}_1 = (x - A - B) - y \cot\alpha \qquad (143)$$

The boundary condition at $y = 0$ gives,

$$\chi_y = - \cot\alpha\, g_o'(Z) + \cot\alpha\, g_o'(\bar{Z})$$
$$- \cot\alpha\, g_1'(\bar{Z}_1) = -1 \qquad (144a)$$

$$\psi_y = -\cot\alpha\, h_o'(Z) + \cot\alpha\, h_o'(\bar{Z}) - \cot\alpha\, h_1'(\bar{Z}_1)$$
$$-i \frac{M}{\cos\alpha} [g_o(Z) - g_o(\bar{Z}) + g_1(\bar{Z}_1)] = i(x - x_o) \qquad (144b)$$

hence,

$$g_1(\bar{Z}_1) = - \bar{Z}_1 \tan\alpha\, e^{i\delta} \qquad (145a)$$

$$h_1(\bar{Z}_1) = i \tan\alpha\, \bar{Z}_1\, e^{i\delta}\{x_o + \frac{\bar{Z}_1}{2} \tan^2\alpha + \frac{Md}{\cos\alpha}\} \qquad (145b)$$

Then,

$$U_{II'} = \tan\alpha\{1-2e^{i\delta} + ik[2e^{i\delta}(x_o+(x-B)\tan^2\alpha$$
$$+ A) - (x_o + x \tan^2\alpha + y \cot\alpha)]\} \qquad (146a)$$

$$V_{II'} = -\tan\alpha\{-e^{i\delta} + ik[2e^{i\delta}(y(\cot\alpha + \frac{M}{\cos\alpha}))$$
$$- (x_o - x + y(\cot\alpha + \frac{M}{\cos\alpha}))]\} \qquad (146b)$$

$$C_{II'} = -\tan\alpha\{ 1-2e^{i\delta} + ik[2e^{i\delta}(x_o+(x-B)(\tan^2\alpha-1)$$
$$+ 2A) - (x_o + x(\tan^2\alpha-1) + 2y \cot\alpha)]\} \qquad (146c)$$

Because of the algebraic complexity of further analysis, additional zone calculations will be made with the assumption that the blades are oscillating in phase, i.e., $\delta = 0$.

For $0.33 \leq A \leq 0.5$ in addition to Zones I, I', II, and II', Zones III and III' will be present in the flow field. In Zone III the perturbation potential is due to the initial waves from Zones I and I' plus the reflected waves at Zones II, II', and III,

$$\begin{aligned}\phi = {} & g_o(Z)+ g_o(\bar{Z})+ g_1(Z_1)+ g_1(\bar{Z}_1)+ g_2(Z_2)+ \\ & k\{h_o(Z)+ h_o(\bar{Z})+ h_1(Z_1)+ h_1(\bar{Z}_1)+ h_2(Z_2) \\ & -i \frac{My}{\cos\alpha}[g_o(Z)- g_o(\bar{Z})- g_1(Z_1)+ g_1(\bar{Z}_1)+ g_2(Z_2)]\} \qquad (147)\end{aligned}$$

where Z, $\bar{Z}$, Z_1, $\bar{Z}_1$ are defined as before, and,

$$Z_2 = x - 2A - y \cot\alpha \qquad (148)$$

The boundary condition at $y = 0$ gives, after considerable algebraic manipulation,

$$g_2 = Z_2 \tan\alpha \qquad (149a)$$

$$h_2 = - \tan\alpha\, Z_2 \{x_o + \frac{Z_2}{2} \tan^2\alpha + 2 \frac{Md}{\cos\alpha}\} \qquad (149b)$$

Then,

$$U_{III} = \tan\alpha\{1- ik[x_o+(x-2B)\tan^2\alpha + 2A+ y \cot\alpha]\} \qquad (150a)$$

$$V_{III} =-\tan\alpha\{1- ik[x_o- x + y(\cot\alpha + \frac{M}{\cos\alpha})]\} \qquad (150b)$$

$$C_{III} =-\tan\alpha\{1- ik[x_o+(x-2B)(\tan^2\alpha-1)+4A +2y \cot\alpha]\} \qquad (150c)$$

In Zone III' the perturbation potential is due to initial waves from Zones I and I' plus the reflected waves at Zones II, II', and III',

$$\phi = g_o(Z) + g_o(\bar{Z}) + g_1(Z_1) + g_1(\bar{Z}_1) + g_2(\bar{Z}_2)$$
$$+ k\{h_o(Z) + h_o(\bar{Z}) + h_1(Z_1) + h_1(\bar{Z}_1) + h_2(\bar{Z}_2)$$
$$- i \frac{My}{\cos\alpha}[g_o(Z) - g_o(\bar{Z}) - g_1(Z_1) + g_1(\bar{Z}_1) - g_2(\bar{Z}_2)]\} \qquad (151)$$

where Z, $\bar{Z}$, Z_1, and $\bar{Z}_1$ are defined as before, and

$$\bar{Z}_2 = x - B - 3A + y \cot\alpha \qquad (152)$$

The boundary condition at y = d gives, after considerable algebraic manipulation,

$$g_2(\bar{Z}_2) = - \bar{Z}_2 \tan\alpha \qquad (153a)$$

$$h_2(\bar{Z}_2) = i \tan\alpha\, \bar{Z}_2 \{x_o + \frac{\bar{Z}_2}{2} \tan^2\alpha + 3 \frac{Md}{\cos\alpha}\} \qquad (153b)$$

Then,

$$U_{III'} = - \tan\alpha\{1 - ik[x_o + (x-3B)\tan^2\alpha + 3A - y \cot\alpha]\} \qquad (154a)$$

$$V_{III'} = - \tan\alpha\{1 - ik[x_o - (x-B) + (d-y)(\cot\alpha + \frac{M}{\cos\alpha})]\} \qquad (154b)$$

$$C_{III'} = \tan\alpha\{1 - ik[x_o + (x-3B)(\tan^2\alpha - 1) + 6A - 2y \cot\alpha]\} \qquad (154c)$$

For 0.25 ≤ A ≤ 0.33, in addition to Zones I, I', II, II', III, and III', Zones IV and IV' will be present in the flow field. In Zone IV the perturbation potential is due to the initial waves from Zones I and I' plus the reflected waves at Zones II, II', III, III',

$$\phi = g_o(Z) + g_o(\bar{Z}) + g_1(Z_1) + g_1(\bar{Z}_1) + g_2(Z_2) + g_2(\bar{Z}_2) + g_3(Z_3)$$
$$+ k\{h_o(Z) + h_o(\bar{Z}) + h_1(Z_1) + h_1(\bar{Z}_1) + h_2(Z_2) + h_2(\bar{Z}_2) + h_3(Z_3)$$
$$-i \frac{My}{\cos\alpha}[g_o(Z) - g_o(\bar{Z}) - g_1(Z_1) + g_1(\bar{Z}_1) + g_2(Z_2) - g_2(\bar{Z}_2) - g_3(Z_3)]\}$$
$$(155)$$

where Z through $\bar{Z}_2$ are defined as before, and

$$Z_3 = x - 4A + y \cot\alpha \qquad (156)$$

The boundary condition at y = d gives after considerable algebraic manipulation,

$$g_3(Z_3) = Z_3 \tan\alpha \qquad (157a)$$

$$h_3(Z_3) = -i \tan\alpha\, Z_3\{x_o + \frac{Z_3}{2}\tan^2\alpha + 4\frac{Md}{\cos\alpha}\} \qquad (157b)$$

Then,

$$U_{IV} = \tan\alpha\{1-ik[x_o+(x+3B)\tan^2\alpha + 3A + y\cot\alpha]\} \qquad (158a)$$

$$V_{IV} = -\tan\alpha\{1-ik[x_o-(x-B)+(y-d)(\cot\alpha+\frac{M}{\cos\alpha})]\} \qquad (158b)$$

$$C_{IV} = -\tan\alpha\{1-ik[x_o+(x+3B)(\tan^2\alpha-1)+6A+2y\cot\alpha]\} \qquad (158c)$$

In Zone IV' the perturbation potential is due to the initial waves from Zones I and I' plus the reflected waves at Zones II, II', III, III', and IV':

$$\begin{aligned}\phi = {} & g_o(Z)+ g_o(\bar{Z})+ g_1(Z_1)+ g_1(\bar{Z}_1)+ g_2(Z_2)+ g_2(\bar{Z}_2)+ g_3(\bar{Z}_3) \\ & +k\{h_o(Z)+ h_o(\bar{Z})+ h_1(Z_1)+ h_1(\bar{Z}_1)+ h_2(Z_2)+ h_2(\bar{Z}_2)+ h_3(\bar{Z}_3) \\ & - \frac{My}{\cos\alpha}[g_o(Z)-g_o(\bar{Z})-g_1(Z_1)+g_1(\bar{Z}_1)+g_2(Z_2) \\ & -g_2(\bar{Z}_2)+ g_3(\bar{Z}_3)]\}\end{aligned} \qquad (159)$$

where Z through $\bar{Z}_2$ are defined as before, and,

$$\bar{Z}_3 = x - B - 3A - y\cot\alpha \qquad (160)$$

The boundary condition at y = 0 gives,

$$g_3(\bar{Z}_3) = -\bar{Z}_3\tan\alpha \qquad (161a)$$

$$h_3(\bar{Z}_3) = i\tan\alpha\,\bar{Z}_3\{x_o+\frac{\bar{Z}_3}{2}\tan^2\alpha + 3\frac{Md}{\cos\alpha}\} \qquad (161b)$$

Then, after considerable algebraic manipulation,

$$U_{IV'} = -\tan\alpha\{1-ik[x_o+(x-4B)\tan^2\alpha + 4A - y\cot\alpha]\} \qquad (162a)$$

$$V_{IV'} = -\tan\alpha\{1-ik[x_o- x - y(\cot\alpha + \frac{M}{\cos\alpha})]\} \qquad (162b)$$

$$C_{IV'} = \tan\alpha\{1-ik[x_o-(x-4B)(\tan^2\alpha-1)+8A-2y\cot\alpha]\} \quad (162c)$$

Further zones may be calculated for values of A less than 0.25 in the same manner.

In the Method of Characteristics it was shown that the non-dimensional pressure, P, is a direct function of C. By integrating this pressure over the blade surface, the lift and pitching moment on the blade may be determined. The derivation of the lift and moment equation is given in the Flutter Analysis. In terms of Garrick and Rubinow's (1946) lift and moment coefficients, for the case of in-phase blade oscillation ($\delta = 0$),

$$L = L_3 + i\,L_4 = -\frac{2}{k^2}\{\int_o^1 C(x,0^+)dx - \int_B^{1+B} C(x,d^-)dx\} \quad (163)$$

$$M = M_3 + i\,M_4 = -\frac{4}{k^2}\{\int_o^1 (x-x_o)\ C(x,0^+)dx - \int_B^{1+B} (x-B-x_o)C(x,d^-)dx\} \quad (164)$$

For $0.25 \leq A \leq 0.33$,

$$L_3 = 0$$

$$L_4 = 8\frac{1}{k}\tan\alpha\{d^2+3A^2+B^2(\tan^2\alpha-1)\} \quad (165)$$

$$M_3 = -16\frac{1}{k^2}\tan\alpha\{A^2-B^2\} \quad (166)$$

$$M_4 = -16\frac{1}{k}\tan\alpha\{x_o(d^2+2A^2+B^2\tan^2\alpha)-8A^3-4d^2A+4AB^2\} \quad (167)$$

If the cascade blades are allowed to oscillate at an arbitrary interblade phase angle, the pitching moment determination can be somewhat simplified by integrating in the following manner: For $0.5 \leq A \leq 1.0$,

$$\begin{aligned} M = M_3 + i\,M_4 = -\frac{4}{k^2}\{ & \int_0^{A-B} (x-x_o)[C_I(x,0^+) - \\ & - e^{-i\delta}\, C_{I'}(x+B,d^-)]dx + \int_{A-B}^{A+B} (x-x_o)[C_I(x,0^+) \\ & - e^{-i\delta}\, C_{II}(x+B,d^-)]dx + \int_{A+B}^{1} (x-x_o)[C_{II'}(x,0^+) \\ & - e^{-i\delta}\, C_{II}(x+B,d^-)]dx\} \end{aligned} \qquad (168)$$

This integration gives

$$\begin{aligned} M_3 = {} & 8\,\frac{1}{k^2}\tan\alpha\{\cos\delta[x_o(2-2A)+(A-B)^2-1-2AB] - k\,B\sin\delta \\ & \{2x_o^2 + x_o[2A+2(\tan^2\alpha-1)]-4A^2-(\tan^2\alpha-1) \\ & (A^2+1-\tfrac{1}{3}B^2)\} - x_o + \tfrac{1}{2}\} \end{aligned} \qquad (169a)$$

$$\begin{aligned} M_4 = {} & 8\,\frac{1}{k}\tan\alpha\{\cos\delta[x_o^2(2A-2)+ x_o(2A^2-4A+2 \\ & + (A^2-B^2-1)\tan^2\alpha) - \tfrac{4}{3}A^3 + 2(1-B^2)A - \tfrac{2}{3} \\ & + \tfrac{2}{3}(1-A^3)\tan^2\alpha] - 2\,\tfrac{1}{k}\,B\sin\delta\,[x_o - A] \\ & + x_o^2 - x_o + \tfrac{1}{3} + (\tfrac{1}{2}x_o - \tfrac{1}{3})\tan^2\alpha\} \end{aligned} \qquad (169b)$$

In most airfoil theory, the non-dimensional pitching moment is written as,

$$C_m = \theta_o[c_{m_\theta} + ik\,c_{m_{\dot\theta}}]e^{ikt} \qquad (170)$$

Relating this to Garrick and Rubinow's (1946) moment coefficients gives (where θ_o is one),

$$c_{m_\theta} = -\frac{1}{2}k^2 M_3 \qquad (171a)$$

$$c_{m_{\dot\theta}} = -\frac{1}{2}k\,M_4 \qquad (171b)$$

V. FLUTTER ANALYSIS

Lane (1956) showed that the flutter analysis of a cascade can be made by considering only a single blade of the cascade oscillating in the flow field. In order to compute the non-dimensional lift and moment acting on that blade, the pressure difference over the top and bottom surfaces must be integrated over the blade. Thus,

$$L = \int_0^1 \{P(x,0^+) - P(x,0^-)\}dx \tag{172}$$

$$M = \int_0^1 (x- x_0)\{P(x,0^+)- P(x,0^-)\}dx \tag{173}$$

where P is the non-dimensional pressure as defined by Eq. (114).

Lift here is positive downward as in Garrick and Rubinow (1946). The non-dimensional pressure on the lower surface of this equivalent blade is that of the lower surface of the blade above adjusted for the interblade phase angle and stagger such that,

$$P(x,0^-) = e^{-i\delta} P(x+B,d) \tag{174}$$

The lift and moment on the equivalent blade can also be written

$$L = \int_0^1 P(x,0^+)dx - \int_B^{1+B} e^{-i\delta} P(x,d)dx \tag{175}$$

$$M = \int_0^1 (x-x_0)P(x,0^+)dx - \int_B^{1+B} (x-B-x_0)e^{-i\delta}P(x,d)dx \tag{176}$$

Separating these into their real and imaginary parts and writing the equations in terms of the local sonic velocities gives,

$$L_R = \gamma M^2\{\int_0^1 C_R(x,0^+)dx - \int_B^{1+B}[\cos\delta \cdot C_R(x,d) + \sin\delta \cdot C_I(x,d)]dx\} \tag{177a}$$

$$L_I = \gamma M^2\{\int_0^1 C_I(x,0^+)dx - \int_B^{1+B}[\cos\delta \cdot C_I(x,d) - \sin\delta \cdot C_R(x,d)]dx\} \tag{177b}$$

$$M_R = \gamma M^2\{\int_0^1 (x-x_o)C_R(x,0^+)dx - \int_B^{1+B} (x-B-x_o) \cdot [\cos\delta\, C_R(x,d) + \sin\delta \cdot C_I(x,d)]dx\} \tag{178a}$$

$$M_I = \gamma M^2\{\int_0^1 (x-x_o)C_I(x,0^+)dx - \int_B^{1+B} (x-B-x_o) \cdot [\cos\delta \cdot C_I(x,d) - \sin\delta \cdot C_R(x,d)]dx\} \tag{178b}$$

where C_R and C_I are the real and imaginary components, respectively.

To evaluate these integrals numerically, the trapezoidal rule is used:

$$\int_o^1 f(x)dx = \frac{\Delta x}{2}[f(x)_o + 2f(x)_1 + 2f(x)_2 + 2f(x)_3 + \ldots \ldots + 2f(x)_{n-2} + 2f(x)_{n-1} + f(x)_n] \tag{179}$$

If oscillation only in bending and torsion (pitching and plunging) is considered and Garrick and Rubinow's (1946) method of expressing lift and pitching moment is used,

$$L = -\frac{1}{2}\rho_\infty\, \hat{c}\, u_\infty^2\, k^2 e^{i\omega t}[\frac{2h_o}{\hat{c}}(L_1 + i\,L_2) + \theta_o(L_3 + i\,L_4)] \tag{180}$$

$$M = -\frac{1}{4}\rho_\infty\, \hat{c}^2 u_\infty^2\, k^2 e^{i\omega t}[\frac{2h_o}{\hat{c}}(M_1 + i\,M_2) + \theta_o(M_3 + i\,M_4)] \tag{181}$$

where $k = \frac{\omega\hat{c}}{u_\infty}$ unlike Garrick-Rubinow's, h_o is the vertical equilibrium position of the elastic axis, and θ_o is the

equilibrium angle of attack of the blade. In dimensional form the lift and pitching moment equations are,

$$L = \hat{c}\, p_\infty\, e^{i\omega t} \{[L_R + i\, L_I]_{\substack{\text{Plunge}\\ \text{mode}}} + [L_R + i\, L_I]_{\substack{\text{Pitch}\\ \text{mode}}}\} \tag{182}$$

&

$$M = \hat{c}^2 p_\infty\, e^{i\omega t} \{[M_R + i\, M_I]_{\substack{\text{Plunge}\\ \text{mode}}} + [M_R + i\, M_I]_{\substack{\text{Pitch}\\ \text{mode}}}\} \tag{183}$$

Thus,

$$L_1 + i\, L_2 = - \frac{\hat{c}}{M^2 k^2 \gamma\, h_o} [L_R + i\, L_I]_{\substack{\text{Plunge}\\ \text{mode}}} \tag{184}$$

Separating the real from the imaginary parts of the equation,

$$L_1 = - \frac{\hat{c}}{M^2 k^2 \gamma\, h_o} L_{R\ \substack{\text{Plunge}\\ \text{mode}}} \tag{185}$$

or,

$$L_1 = - \frac{\hat{c}}{k^2\ h_o} \{\int_o^1 C_R(x,0^+)dx - \int_B^{1+B} [\cos\delta \cdot C_R(x,d) + \sin\delta \cdot C_I(x,d)]dx\}_{\text{Plunge mode}} \tag{186a}$$

Likewise,

$$L_2 = - \frac{\hat{c}}{k^2\ h_o} \{\int_o^1 C_I(x,0^+)dx - \int_B^{1+B} [\cos\delta \cdot C_I(x,d) - \sin\delta \cdot C_R\ (x,d)]dx\}_{\text{Plunge mode}} \tag{186b}$$

$$L_3 = -2 \frac{1}{k^2\ \theta_o} \{\int_o^1 C_R(x,0^+)dx - \int_B^{1+B} [\cos\delta \cdot C_R(x,d) + \sin\delta \cdot C_I(x,d)]dx\}_{\text{Pitch mode}} \tag{187a}$$

$$L_4 = -2 \frac{1}{k^2 \theta_o} \{ \int_o^1 C_I(x,0^+)\,dx - \int_B^{1+B} [\cos \delta \cdot C_I(x,d) - \sin \delta \cdot C_R(x,d)]\,dx \} \qquad (187b)$$

Pitch mode

$$M_1 = -2 \frac{\hat{c}}{k^2 h_o} \{ \int_o^1 (x-x_o) C_R(x,0^+)\,dx - \int_B^{1+B} (x-B-x_o) \cdot [\cos \delta \cdot C_R(x,d) + \sin \delta \cdot C_I(x,d)]\,dx \}$$

Plunge mode (188a)

$$M_2 = -2 \frac{\hat{c}}{k^2 h_o} \{ \int_o^1 (x-x_o) C_I(x,0^+)\,dx - \int_B^{1+B} (x-B-x_o) \cdot [\cos \delta \cdot C_I(x,d) - \sin \delta \cdot C_R(x,d)]\,dx \}$$

Plunge mode (188b)

$$M_3 = -4 \frac{1}{k^2 \theta_o} \{ \int_o^1 (x-x_o) C_R(x,0^+)\,dx - \int_B^{1+B} (x-B-x_o) \cdot [\cos \delta \cdot C_R(x,d) + \sin \delta \cdot C_I(x,d)]\,dx \}$$

Pitch mode (189a)

$$M_4 = -4 \frac{1}{k^2 \theta_o} \{ \int_o^1 (x-x_o) C_I(x,0^+)\,dx - \int_B^{1+B} (x-B-x_o) \cdot [\cos \delta \cdot C_I(x,d) - \sin \delta \cdot C_R(x,d)]\,dx \}$$

Pitch mode (189b)

where the integrals are computed using the trapezoidal rule and the two different oscillatory modes are determined by the boundary conditions at the blade surfaces.

Bisplinghoff, Ashley, and Halfman (1955) define flutter as "the dynamic instability of an elastic body in an air-stream." In Chapter 9 they describe the physical nature of the flutter phenomenon. Basically, flutter occurs when the characteristic determinant of the equations of motion of the blade vanishes.

If blade oscillation in two degrees of freedom (pitch and plunge) is considered, the equations of motion of the cascade blade can be written as,

$$M\ddot{h} + S_\theta \ddot{\theta} + C_h h = L \tag{190}$$

$$S_\theta \ddot{h} + I_\theta \ddot{\theta} + C_\theta \theta = M_\theta \tag{191}$$

where the total inertia force acting on the blade is,

$$F_I = -\,(M\ddot{h} + S_\theta \ddot{\theta}) \tag{192}$$

the elastic restoring force is,

$$F_R = -\,C_h h \tag{193}$$

the sum of the moments of the inertia forces about the elastic axis is,

$$M_I = -\,(S_\theta \ddot{h} + I_\theta \ddot{\theta}) \tag{194}$$

and the elastic restoring moment is,

$$M_R = -\,C_\theta \theta \tag{195}$$

L and M_θ are the aerodynamic lift and pitching moment about the elastic axis. With the assumption of simple harmonic oscillation, the blade motion can be written,

$$h = h_o e^{i\omega t} \tag{196a}$$

$$h = -\,\omega^2 h_o e^{i\omega t} \tag{196b}$$

$$\theta = \theta_o\, e^{i\omega t} \tag{197a}$$

$$\ddot{\theta} = -\,\omega^2 \theta_o\, e^{i\omega t} \tag{197b}$$

The equations of motion thus become,

$$L = e^{i\omega t}[-\,\omega^2 M h_o - \omega^2 S_\theta \theta_o + C_h h_o] \tag{198}$$

$$M_\theta = e^{i\omega t}[-\,\omega^2 S_\theta h_o - \omega^2 I_\theta \theta_o + C_\theta \theta_o] \tag{199}$$

Equating these to Garrick-Rubinow's standard lift and moment expressions and following Garrick-Rubinow's (1946) method, the equations can be written as,

$$(\Omega_h X-\mu+ L_1+ i L_2)\frac{2h_o}{\hat{c}} +(-\mu x_\theta+ L_3+ i L_4)\theta_o = 0 \tag{200a}$$

$$(-\mu x_\theta + M_1 + i M_2)\frac{2h_o}{\hat{c}} +(\Omega_\theta X - \mu r_\theta^2 + M_3 + i M_4)\theta_o = 0 \tag{200b}$$

The two simultaneous flutter equations form an eigenvalue problem in $\frac{2h_o}{\hat{c}}$ and θ_o, thus, the determinant of coefficients (the characteristic determinant) must be zero:

$$\begin{vmatrix} \Omega_h X - \mu + L_1 + i L_2 & - \mu x_\theta + L_3 + i L_4 \\ -\mu x_\theta + M_1 + i M_2 & \Omega_\theta X - \mu r_\theta^2 + M_3 + i M_4 \end{vmatrix} = 0 \tag{201}$$

This is the flutter determinant with unknown X. If the determinant is expanded, and the real and imaginary parts equated to zero the result is two equations that given a value of k and M will yield values of X. For a specific Mach number, the flutter point occurs at the value of k that produces identical values of X in the two equations. From this value of k and X the flutter speed and frequency can be calculated.

The two flutter equations are:

(Real part)

$$\Omega_h\Omega_\theta X^2+[\Omega_\theta(L_1-\mu) + \Omega_h(M_3-\mu r_\theta^2)]X + C_R = 0 \tag{202}$$

(Imaginary part)

$$(\Omega_\theta L_2 + \Omega_h M_4)X + C_I = 0 \tag{203}$$

where,

$$C_R = \mu[x_\theta(M_1 + L_3) - (M_3 - \mu r_\theta^2) - L_1 r_\theta^2 - \mu x_\theta^2] + D_R \qquad (204a)$$

$$C_I = \mu[x_\theta(M_2 + L_4) - M_4 - L_2 r_\theta^2] + D_I \qquad (204b)$$

and where,

$$D_R = L_1M_3 - L_3M_1 - L_2M_4 + L_4M_2 \qquad (205a)$$

$$D_I = L_1M_4 - L_4M_1 + L_2M_3 - L_3M_2 \qquad (205b)$$

Solving for X,

(Real Part)

$$X = \frac{-[\Omega_\theta(L_1-\mu)+\Omega_h(M_3-\mu r_\theta^2)] \pm \sqrt{[\Omega_\theta(L_1-\mu)+\Omega_h(M_3-\mu r_\theta^2]^2 - 4C_R\Omega_h\Omega_\theta}}{2\Omega_h\Omega_\theta} \qquad (206)$$

(Imaginary Part)

$$X = \frac{-\,C_I}{\Omega_\theta\, L_2 + \Omega_h\, M_4} \qquad (207)$$

The solution to the flutter problem is then to obtain values of X as a function of k until the real and imaginary solutions converge. At this "flutter point' the non-dimensional flutter frequency is,

$$\frac{\omega_F}{\omega_\theta} = \frac{1}{\sqrt{X}} \qquad (208)$$

and the non-dimensional freestream flutter speed is

$$\frac{U_F}{\hat{c}\omega_\theta} = \frac{1}{k\sqrt{X}} \qquad (209)$$

It should be noted that the interblade phase angle is an open parameter (as well as k) in calculating lift and moment on the cascade blade. Lane (1956) showed that the minimum flutter speed of a cascade occurs at a non-zero interblade phase angle and that this is the critical phase angle at

which the cascade will oscillate. For an unstaggered cascade this critical phase angle was shown to be 180°, but for a staggered cascade it can only be found by repeated flutter calculations at various phase angles. The problem is solved when the minimum flutter speed is determined.

If only a single degree of freedom of motion (pitch oscillation) exists, the flutter problem is greatly simplified. The equation of motion of the blade can then be written,

$$I_\theta \ddot{\theta} + (1+ig)C_\theta \theta = M_\theta \qquad (210)$$

Here the sum of the moments of the inertia forces about the elastic axis is,

$$M_I = - I_\theta \ddot{\theta} \qquad (211)$$

the elastic restoring moment is,

$$M_R = - (1+ig)C_\theta \theta \qquad (212)$$

and the aerodynamic pitching moment about the elastic axis is,

$$M_\theta = - \frac{1}{4} \rho_\infty \hat{c}^2 u_\infty^2 k^2 \theta_o e^{i\omega t}(M_3 + i M_4) \qquad (213)$$

With the assumption of simple harmonic oscillation as before, the blade motion can be written as,

$$\theta = \theta_o e^{i\omega t} \qquad (214)$$

$$\ddot{\theta} = -\omega^2 \theta_o e^{i\omega t} \qquad (215)$$

The equation of motion then becomes,

$$e^{i\omega t}[-I_\theta \omega^2 \theta_o + (1+ig)C_\theta \theta_o] = - \frac{1}{4}\rho_\infty \hat{c}^2 u_\infty^2 k^2 \theta_o e^{i\omega t}(M_3 + i M_4) \qquad (216)$$

Separating the equation into its real and imaginary parts gives,

$$-I_\theta\omega^2 + C_\theta + \frac{1}{4}\rho_\infty\hat{c}^4\omega^2M_3 = 0 \tag{217a}$$

$$g\ C_\theta + \frac{1}{4}\rho_\infty\hat{c}^4\omega^2M_4 = 0 \tag{217b}$$

or as in Garrick and Rubinow (1946)

$$\Omega_\theta X - \mu r_\theta^2 + M_3 = 0 \tag{218a}$$

$$M_4 + g\ \Omega_\theta X = 0 \tag{218b}$$

Thus, the solution to the single-degree-of-freedom problem is to obtain values of M_4 as a function of k until the second equation is satisfied. The value of M_3 for that k is then used in the first equation to determine if X is reasonable:

$$X = 1 - \frac{M_3}{\Omega_\alpha} > 0 \tag{219}$$

The non-dimensional flutter frequency is as before,

$$\frac{\omega_F}{\omega_\theta} = \frac{1}{\sqrt{X}} \tag{220}$$

and the non-dimensional freestream flutter speed is,

$$\frac{U_F}{\hat{c}\omega_\theta} = \frac{1}{k\sqrt{X}} \tag{221}$$

Again it must be noted that in the case of staggered cascades the flutter computations must be repeated with varying interblade phase angle until the minimum flutter speed is found. This will give the critical flutter condition.

As in the analysis of the single airfoil, in the absence of structural damping, aerodynamic instability (flutter) occurs when $M_4 \leq 0$. When M_4 is written as the pitch damping

derivative, Eq. (171b), it should be noted that aerodynamic undamping occurs when the pitch damping derivative is positive.

VI. COMPUTATIONAL PROCEDURE

The computer programs used to calculate the cascade flow field are based on one used by Platzer and Pierce (1970) to calculate pressure distributions on an airfoil oscillating with wind tunnel interference. The programs here, however, in addition to calculating the pressure distribution on the blade of a cascade integrate the pressure distribution over the blade surface and calculate the flutter frequency and freestream flutter velocity. One program (Program A) is for two-degree-of-freedom (pitch and plunge) flutter calculations while the second (Program B), merely a simplification of the first, is for single-degree-of-freedom (pitch) flutter calculations.

Both programs are valid only for cascades with supersonic leading-edge locus vibrating with small amplitude oscillations. The programs calculate the flow field between two adjacent blades using the method of characteristics finite difference equations developed previously and integrate the pressure distributions on the blade surfaces using the trapezoidal rule.

The computational molecules shown in Figures 2, 3, and 4 are used to calculate the flow field quantities at P_{22} in the general flow field, on the lower blade, and on the upper blade, respectively. The distance Δx shown in the figures is determined by,

$$\Delta x = \frac{2A}{\nu} \tag{222}$$

where ν is the grid fineness ratio. This parameter is an arbitrary input variable into the programs and is equal to one less than the total number of grid points on any Mach line that runs from one blade to the other.

Input parameters are entered into the programs by means of the NAMELIST option of FORTRAN. In using this option, input parameters are combined in a titled list and then referred to in the programs by this title. In both Program A and B the NAMELIST title is the same (NAM1); however, the input variables in each are slightly different. The real advantage of using NAMELIST is that each data card need only contain the value of one input variable while the format of the data card is such that the variable's name is punched on the card as well as its value. In this way, when a different value for an input variable is desired, rather than changing the entire data deck, only one easily identifiable data card need be changed.

The format for a NAMELIST data deck is as follows: The first column in each card of the deck is left blank. The first card starts in the second column with the symbol & (ampersand) followed immediately by the title of the NAMELIST (here, NAM1). The next cards list the variables and their input values, one per card, in any order. The format here, starting in the second column and followed by a comma, is: Variable Name = Value. The last card in the data deck and

the signal to the computer that the NAMELIST has ended again starts in the second column with the symbol & (ampersand) followed immediately by END.

The value of an input variable may be written in any format as long as integer variables are written without decimal points and real variables are written with a decimal point. Blanks are taken as zeroes, but a comma must appear somewhere between a desired input value and the next data card.

In both programs, prior to the NAMELIST dataset, the date is entered on a data card in the first twelve spaces.

To illustrate the input procedures, the following two examples are proposed:

Two Degree of Freedom Flutter

$\nu = 100$
$M = \sqrt{2}$
$\gamma = 1.4$
$k = 2$
$x_0 = 0.5$
$d = 0.6$
$\beta = 26.565°$
$\delta = 180°$
$\mu = 500$
$r_\theta = 0.5$
$x_\theta = 0.1$
$\frac{\omega_h}{\omega_\theta} = 1.5$
$\Delta 2/k = 1$
$\hat{c} = 1.2$

One Degree of Freedom Flutter

$\nu = 100$
$M = \sqrt{2}$
$\gamma = 1.4$
$k = 2$
$x_0 = 0.5$
$d = 0.6$
$\beta = 26.565°$
$\delta = 180°$
$\mu = 500$
$r_\theta = 0.5$
$g = 2\%$
$\Delta 2/k = 1$
$\hat{c} = 1.2$

The data cards needed are as follows:

Card	Program	Format	Description
1	A & B	16 JUNE 1972	Date
2	A & B	&NAM1	NAMELIST title
3	A & B	NGRDFN = 100,	Grid fineness ratio, must be an even integer, less than 400.
4	A & B	FSTRMN = 1.414214,	Freestream Mach number
5	A & B	RTOSPH = 1.4,	Ratio of specific heats
6	A & B	REDFRQ = 2.0,	Reduced frequency
7	A & B	XSUBO = 0.5,	Elastic axis position
8	A & B	TNWDST = 0.6,	Blade distance
9	A & B	STGANG = 26.565,	Stagger angle, $\tan \beta \leq \cot \alpha$.
10	A & B	FAZE = 180.0,	Interblade phase angle
11	A & B	MUU = 500.0,	Blade density parameter
12	A & B	RSUBA = 0.5,	Radius of gyration
13	A	XSUBA = 0.1,	Distance from the blade elastic axis to its center of gravity.
13	B	GSUBA = 0.02,	Structural damping coefficient
14	A & B	INCRE = 1.0,	2/k increment
15	A & B	FIN = 1.2,	Chord length, $\hat{c} \geq 2A$.
16	A	HAFREQ = 1.5,	Ratio of natural frequencies, bending to torsion.
16	B	&END	End of NAMELIST
17	A	&END	End of NAMELIST

The definitions of these parameters are given again in the alphabetical listing of all program variables in Appendix A.

Program A has eight subroutines. They are as follows:

INPUT	Reads in all input data and sets up the finite difference grid.
INTIAL	Initializes all the flow quantities at the leading-edge, starts the integration, and initializes most of the logic variables.
MACHLN	Computes the flow quantities along the initial left and right-running Mach lines at the given grid point.
HIFOIL	Computes the flow quantities on the lower surface of the upper airfoil at the given grid point, and continues the integration of the pressure there.
GENFPT	Computes the flow quantities at a general flow field grid point.
LOFOIL	Computes the flow quantities on the upper surface of the lower airfoil at the given grid point, and continues the integration of the pressure there.
COMPXY	Computes the x and y position of the given grid point.
FLUTER	Computes the oscillation frequency ratio, X, for both real and imaginary flutter equations.

Program B contains all of these subroutines except FLUTER. Since the flutter problem is greatly simplified in the single degree of freedom case, the flutter calculations are made in the main program.

Program A contains at least two "DO" loops. The inner one is for calculating the flow field with first pitch oscillation boundary conditions and then plunge oscillation boundary conditions. This loop is not present in Program B

since only the flow field due to pitch oscillation is desired. The next outer loop is used to increment 2/k (1/k in Garrick-Rubinow's (1946) notation) starting at that reduced frequency specified in INPUT. This loop is continued until hopefully the flutter point is found. Any loops outside of these are used to make multiple flutter calculations in one run with varying input parameters.

In INPUT the grid is determined using the grid fineness ratio, blade spacing, and Mach number. The value of Δx as shown in Eqs. (96) is obtained. The stagger angle is then adjusted such that the distance the upper blade is staggered back is the nearest multiple number of Δx increments. The "Compatible Stagger Angle" is then printed out. The upper blade may be staggered back such that the distance between its leading-edge and its intersection with the initial left-running Mach line from the lower blade is zero, but no further. With less stagger this distance must be greater than or equal to $2\Delta x$, or,

$$A - B \geq 2\Delta x \tag{223}$$

This allows a minimum of three grid points in the primed flow field zones as shown in Figure 5. This is the minimum number of points required when the value of the last point must be linearly extrapolated from the other two, as is the procedure in the programs.

Flow quantities along the initial left and right-running Mach lines are computed using the equations derived in the Method of Characteristics for initial conditions. Those

quantities along the blade surfaces are computed using the equations derived for the appropriate boundary condition. The values of the normal velocities are derived as a result of the analysis of the flow tangency condition and have been given previously.

Because the programs compute the flow field in rows of right-running Mach lines starting with points on the initial left-running Mach line, provision must be made for the computation of points in the area referred to as Zone I'. The reason is that once the computation reaches the intersection of the initial left-running Mach line from the lower blade and the initial right-running Mach line from the upper blade, the points on the initial left-running Mach line become general flow field points requiring those equations for the flow quantity computation. To use those equations the flow quantities along the last left-running Mach line in Zone I' must be known. They are obtained in two ways depending on the value of the interblade phase angle: If the phase angle is non-zero, the entire flow field in Zone I' (along with the integration over the bottom surface of the top blade) is computed concurrently with the computation of the flow field in Zone I. If the phase angle is zero, only that row in Zone I that is the image of the last left-running Mach line in Zone I' must be determined since the flow quantities of these two rows are the same (except for the signs of U and C).

The pressure distributions on the surfaces of the blades are integrated point by point as they are being calculate

Since the pressure coefficient at the last point in each zone of influence is unknown, its value must be linearly extrapolated from the preceding points. The pressure distribution of the upper blade is adjusted so as to represent the pressure distribution on the bottom surface of the lower blade:

$$C(x,0^-) = e^{-i\delta} C(x+B,d) \tag{224}$$

In this way the result of the integration is the lift and pitching moment on a representative blade of the cascade.

Termination of the flow field calculation starts when the distance from the leading-edge of the lower blade to the point of calculation on the lower blade reaches the prescribed chord length. Because HIFOIL must be called at least once for proper program termination, this chord length has a lower bound:

$$\hat{c} \geq 2A \tag{225}$$

Since the flow field is being calculated along rows of right-running Mach lines, the last zone along the upper blade is then computed to complete the calculation. Although the flow field is terminated when the calculation reaches the prescribed chord length, the integration of the pressure distributions is terminated when the chord length reaches unity.

When the values of total lift and pitching moment have been determined, Program A enters them in the flutter equations, Eq. (206) and Eq. (207) via subroutine FLUTER. Program B, computing only the pitching moment, enters these values in Eqs. (218). Convergence to the flutter point is then determined.

In Program A, the difference between solutions to the real flutter equation and that to the imaginary flutter equation is noted after each computation. If the absolute value of the difference is less than 10^{-6}, the flutter point is considered to be determined. When the difference changes sign, indicating convergence somewhere in the frequency interval, the 2/k increment is divided by five and the computation is started again at a new value of 2/k determined by adding the new increment to the previous lower value. Each time the flutter point is passed the increment is made smaller. The flutter point is considered to be ascertained when the increment is less than or equal to 10^{-5}.

In Program B, the value of X is determined from Eq. (218a). This value is then used along with M_4 to determine the value of Eq. (218b). If the absolute value of Eq. (218b) is less then 10^{-6}, the flutter point is considered to be determined. If Eq. (218b) is negative, indicating convergence somewhere in the frequency interval, the same procedure is followed as in Program A. When the flutter point is determined, the value of X is tested to insure a realistic flutter frequency as shown in Eq. (219).

The output from Program A consists of the input parameters (including such items as x and the compatible stagger angle); the values for X_R and X_I as defined in Eq. (206) and Eq. (207), respectively, for each reduced frequency calculated; and the non-dimensional flutter frequency and flutter speed as defined in Eq. (208) and Eq. (209), respectively. The value of

"1/k" given in the output corresponds to the reciprocal of the reduced frequency as defined by Garrick and Rubinow (1946). This is equivalent to our 2/k.

The output from Program B is identical to that of Program A except that Program B gives the values of M_4, M_3, X, and the system damping $(M_4 - g\Omega_\theta X)$ as shown in Eq. (218b) for each reduced frequency calculated.

A flow diagram of Program A is given in Appendix B.

VII. RESULTS

A. COMPARISON OF ELEMENTARY THEORY WITH MILES

Miles (1956) presented an analysis of supersonic flow past an oscillating airfoil subjected to wall interference in a solid-wall wind tunnel. Using Laplace transform techniques he derived the following results for the moment coefficient (his Eq. (7.18))

$$c_{m\alpha} = 4\tan\alpha\{(2N+1)[x_o - \tfrac{1}{2}] - N(N+1)Ax_o + \tfrac{1}{6}N(N+1)(2N+1)A^2\} \quad (226a)$$

$$c_{mq} = -4\tan\alpha\{(2N+1)[x_o^2 - x_o + \tfrac{1}{3}] - N(N+1)A\,[x_o^2 - x_o + \tfrac{1}{2}] + \tfrac{1}{12}N^2(N+1)^2A^3\} \quad (226b)$$

$$c_{m\dot{\alpha}} = 4\tan\alpha\{-(2N+1)\tan^2\alpha[\tfrac{1}{2}x_o - \tfrac{1}{3}] - N(N+1)A\,[x_o - \tfrac{1}{2}] + \tfrac{1}{6}N(N+1)(2N+1)(\tan^2\alpha + 2)\cdot A^2x_o - \tfrac{1}{12}N^2(N+1)^2(2\tan^2\alpha + 3)A^3\} \quad (226c)$$

where N is the largest integer smaller than A^{-1} and α, $\dot{\alpha}$ subscripts represent angle of attack. Here, for a slowly oscillating airfoil,

$$C_m = \alpha_o[c_{m\alpha} + ik(c_{mq} + c_{m\dot{\alpha}})]e^{ikt} \quad (227)$$

unlike Eq. (170).

As stated by Miles, this case is identical to that of the unstaggered cascade oscillating with an interblade phase angle of 180°, and the values of pitching moment as defined in Eq. (170) and Eq. (171) based on our elementary theory for $B = 0$, $\delta = 180$, and $0.5 \leq A \leq 1.0$, prove to be the same as those shown in Eqs. (226) with $N = 1$. The torsional

stability boundaries $c_{m\dot{\theta}} = 0$ for an unstaggered cascade oscillating with an interblade phase angle of 180° as computed from Eq. (171b) are shown in Figure 6 for various values of solidity.

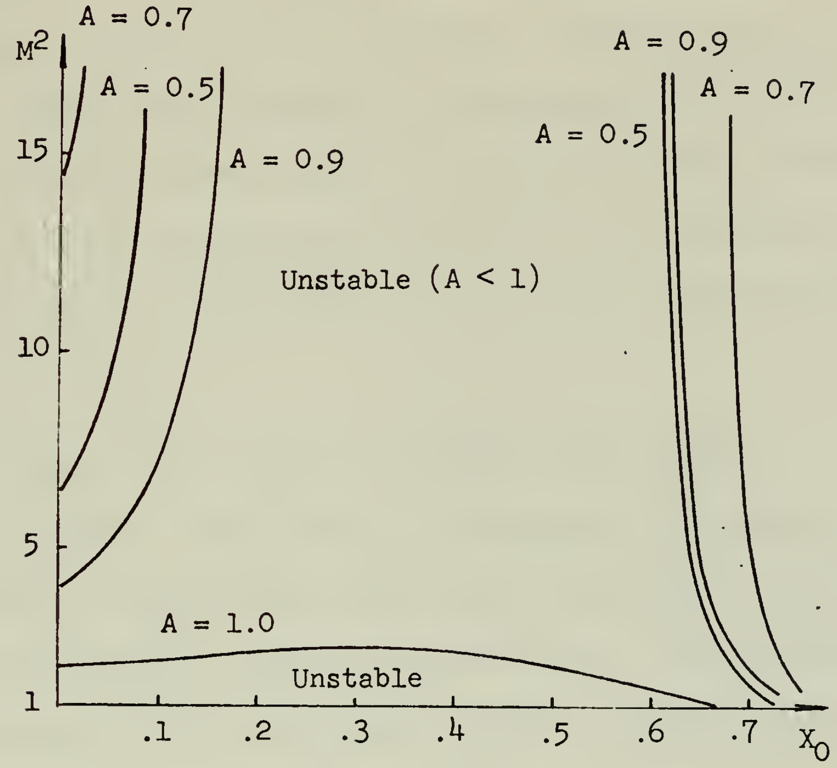


Figure 6

The large destabilizing influence of interference causing instability over the whole supersonic Mach number range for a wide range of elastic axis positions can be seen. Also shown is the interference - free boundary (A = 1) for the single airfoil in an unbounded supersonic flow (Garrick and Rubinow 1946) exhibiting instability as is well known only for low supersonic Mach numbers. As demonstrated above,

these stability boundaries also apply to the slowly oscillating airfoil subjected to solid-wall wind tunnel interference.

B. COMPARISON OF ELEMENTARY THEORY WITH LANE

Lane (1957) extended Miles' Laplace transform solution to the analysis of oscillating cascades with supersonic leading-edge locus. By expanding the blade pressure distributions (his Eq. (13)) with respect to frequency and retaining only terms up to the first power in frequency, the pressure jump across a slowly oscillating blade can be obtained. This pressure jump is identical to that being integrated in Eq. (168).

C. COMPARISON OF ELEMENTARY THEORY WITH DRAKE

Drake (1956) using Laplace transform techniques presented an analysis of supersonic flow past an oscillating airfoil in a wind tunnel with free jet boundaries. His pitching moment coefficient is defined in the English system, equivalent to Eq. (170), is,

$$C_m = \alpha_o [m_\alpha + ik\, m_{\dot{\alpha}}]\, e^{ikt} \tag{228}$$

for a slowly oscillating airfoil where α again refers to angle of attack. Here,

$$m_\alpha = 4 \tan \alpha [(x_o - 1) h_1(A) + h_2(A)] \tag{229a}$$

$$m_{\dot{\alpha}} = 4 \tan \alpha \{h_3(A) - h_2(A) - x_o (x_o - 1) h_1(A) + \tan^2 \alpha \left(M^2 A \frac{\partial}{\partial A} - 1\right) [\{x_o - 1\} h_2(A) + h_3(A)]\} \tag{229b}$$

where for $0.5 \leq A \leq 1.0$,

$$h_1(A) = 2A - 1 \tag{230a}$$

$$h_2(A) = -A^2 + 2A - \frac{1}{2} \tag{230b}$$

$$h_3(A) = \frac{1}{3}A^3 - A^2 + A - \frac{1}{6} \tag{230c}$$

This case is identical to that of the unstaggered cascade oscillating in phase ($\delta = 0$), and the values of pitching moment as defined in Eq. (170) and Eq. (171) based on our elementary theory for $\delta = 0$, $\beta = 0$, and $0.5 \leq A \leq 1.0$ prove to be the same as those shown in Eq. (229).

Drake (1957) extended his work to the case of the airfoil oscillating in a wind tunnel with porous walls. Platzer (1971) will publish an analysis of this problem using our elementary theory. His pitch damping coefficient as defined in Eq. (170), for a slowly oscillating airfoil is,

$$\begin{aligned} c_{m\dot{\theta}} = &- 4 \tan \alpha \{x_o^2 + x_o(\frac{1}{2} \tan^2\alpha - 1) + \frac{1}{3}(1 - \tan^2\alpha) \\ &+ \frac{2(1-A)}{1+\sigma\tan\alpha} \{[1-\sigma \tan \alpha)(1-\tan^2\alpha) + \frac{2\sigma \tan^3\alpha}{1+\sigma\tan\alpha}] \\ &[X] + [(1-\sigma \tan \alpha)(x_o + A(1+\tan^2\alpha))][Y]\}\} \end{aligned} \tag{231}$$

where

$$X = (1-A)[-\frac{1}{2}x_o + \frac{1}{6}(A+2]$$

$$Y = x_o - \frac{1}{2}(A+1)$$

and $0.5 \leq A \leq 1.0$. Here $0 \leq \sigma \leq \infty$ where $\sigma = 0$ represents the solid-wall wind tunnel case and $\sigma = \infty$ represents a free-jet wind tunnel boundary. With these limiting values of porosity substituted in Eq. (231) the comparison with Eq. (171b) is exact for the case of $\delta = 180$ and $\delta = 0$, respectively.

Further the results check with Drake (1957). Stability boundaries for an airfoil oscillating slowly in a porous-wall wind tunnel are given in Figure 7 for a tunnel aspect ratio, A = 0.5 and for various values of tunnel porosity. An increase in tunnel porosity is seen to have a large stabilizing influence.

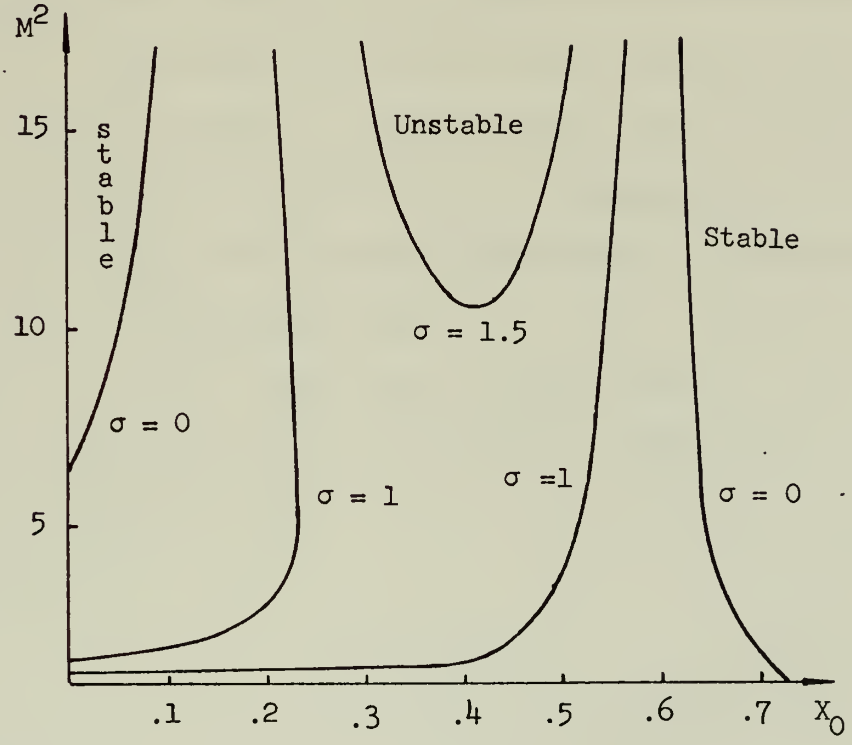


Figure 7

D. AIRFOIL - AIRFOIL INTERFERENCE

Our elementary solution of the slowly oscillating cascade problem can be used to study other interference problems as well. The interference between two airfoils may be of interest for certain proposed space shuttle configurations, although two-dimensional flow can only give an indication of

trends. Our elementary solution applied to the unstaggered cascade oscillating with an interblade phase angle of 180°, as stated before, is equivalent to an airfoil subjected to solid-wall wind tunnel interference. If the upper wind tunnel wall is considered a large stationary airfoil (idealized as a flat plate at zero angle of attack), the theory can be used to determine the non-dimensional pressure distribution along the upper surface of a smaller airfoil mounted closely below it. Using the non-interference pressure distribution over the lower surface of the airfoil, the net pressure distribution over the airfoil can be integrated over the chord length and the pitch damping coefficient as defined in Eq. (171b) can be determined as,

$$\begin{aligned} c_{m\dot{\theta}} = -2 \tan\alpha \,\{ & x_o^2(4-2A) - x_o[3A^2 - 4A + 2 \\ & + (A^2 - 2)(\tan^2\alpha - 1)] + 2A^3 - 2A - \frac{2}{3}(2-A^3) \\ & (\tan^2\alpha - 1)\} \end{aligned} \qquad (232)$$

where

$$A = 2h \cot\alpha$$

h = airfoil distance

and $0.5 \leq A \leq 1.0$.

The stability boundaries for an airfoil undergoing interference of this type are shown in Figure 8. The effect of interference is strongly destabilizing. This case further applies to an airfoil mounted close to only one solid wall in a wind tunnel.

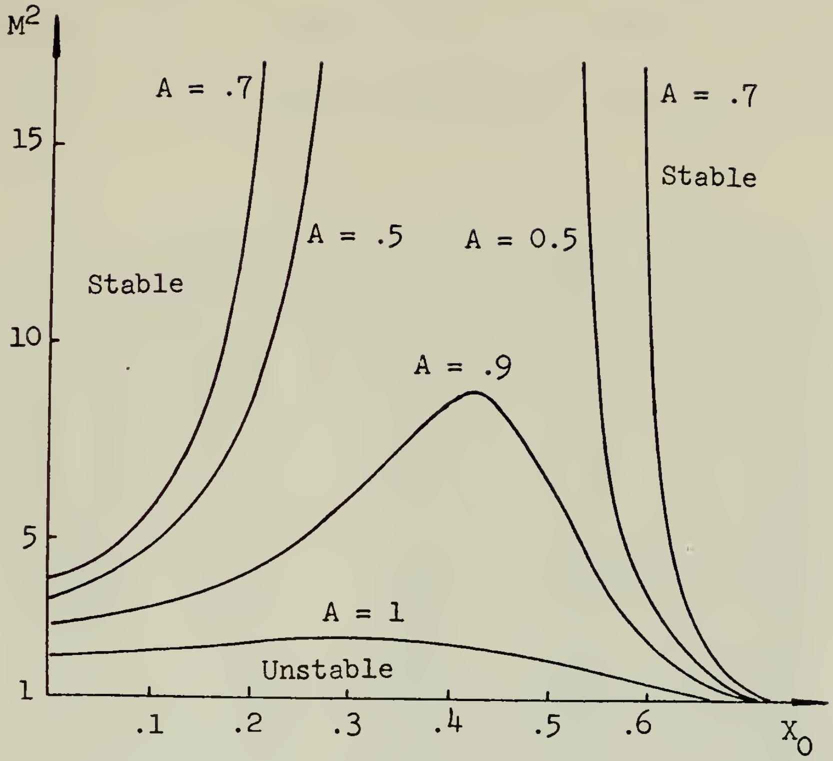


Figure 8

E. COMPARISON OF THE METHOD OF CHARACTERISTICS WITH ELEMENTARY THEORY

Values of flow field quantities, U, V, and C, both real and imaginary, based on computations using Program B were compared with those quantities using the elementary solution as applied to cascades. Comparisons were made for various values of A, stagger, and interblade phase angle. Points along both surfaces of the blades as well as in the flow field were compared. In the case of zero interblade phase angle, values of A as low as 0.25 were compared. The restriction of supersonic leading-edge locus was maintained. All

values compared were within 2% up to reduced frequencies of 0.1. Differences between 1% and 2% were obtained only at Mach numbers near one ($M = \sqrt{1.25}$) at a reduced frequency of 0.1. A sample of the flow quantities compared is given in Table 1.

#	x	y	C22R (M.O.C.)	C22R (S.S.C.)	C22I (M.O.C.)	C22I (S.S.C.)
1	0.10	0.00	-1.000	-1.000	0.000	0.000
	0.50	0.00	0.659	0.660	1.120	1.120
	0.60	0.00	0.659	0.660	1.120	1.120
	0.10	0.50	0.800	0.800	0.600	0.600
	0.50	0.50	-1.197	-1.200	0.699	0.700
	0.60	0.50	-1.197	-1.200	0.699	0.700
2	0.20	0.00	-1.000	-1.000	0.000	0.000
	0.50	0.00	0.659	0.660	1.119	1.120
	0.60	0.00	0.659	0.660	1.119	1.120
	0.44	0.50	0.800	0.800	0.600	0.600
	0.74	0.50	-1.197	-1.200	0.699	0.700
	0.84	0.50	-1.197	-1.200	0.699	0.700
3	0.20	0.00	-1.997	-2.000	0.060	0.060
	0.50	0.00	-1.984	-2.000	0.149	0.150
	0.74	0.00	1.510	1.500	2.203	2.222
	0.80	0.00	1.530	1.522	2.187	2.211
	0.40	1.00	1.627	1.629	1.161	1.162
	0.50	1.00	-2.322	-2.353	1.633	1.650
	0.74	1.00	-2.252	-2.264	1.708	1.738

#1	#2	#3
$M = \sqrt{2}$	$M = \sqrt{2}$	$M = \sqrt{1.25}$
$A = 0.5$	$A = 0.5$	$A = 0.5$
$\beta = 0.0$	$\beta = 25.64$	$\beta = 13.50$
$\delta = 36.87$	$\delta = 36.87$	$\delta = 36.87$
$x_o = 0.0$	$x_o = 0.0$	$x_o = 0.0$

Table 1

Further, comparisons of the out of phase pressure distribution on the upper surface of a staggered cascade blade in supersonic flow of $M = \sqrt{1.25}$ and $M = \sqrt{2}$ as predicted by the method of characteristics and the elementary theory are shown in Figure 9 ($M = \sqrt{2}$, $B = .12$, $x_0 = 0$, $\delta = 0$) and Figure 10 ($M = \sqrt{1.25}$, $B = .12$, $x_0 = 0$, $\delta = 0$). Here, the solid lines indicate the elementary theory, and the dashed lines indicate the method of characteristics.

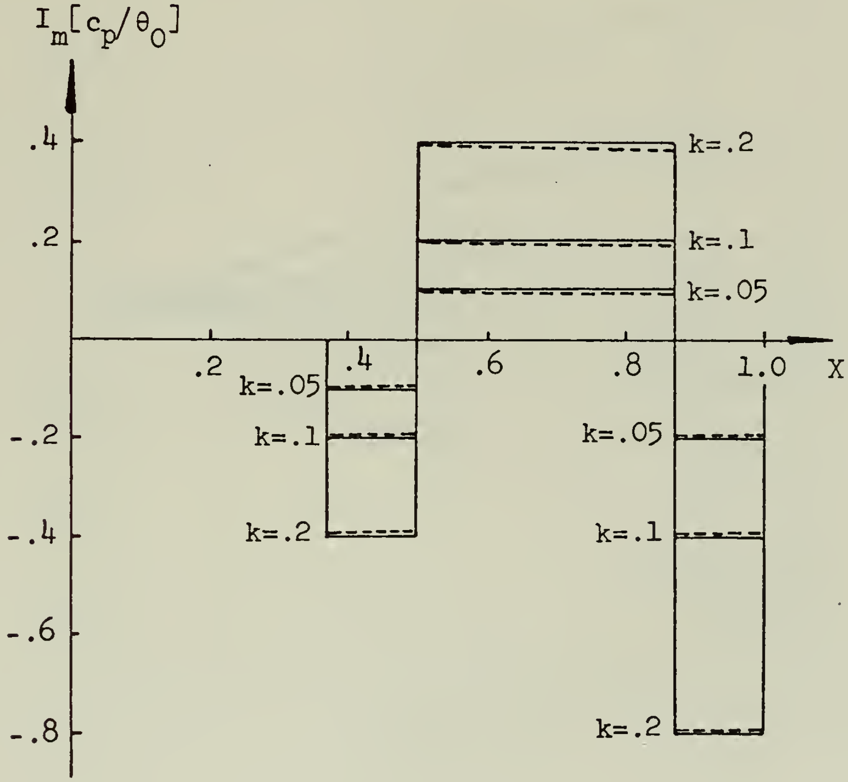


Figure 9

Agreement between the two methods is quite good for reduced frequencies up to 0.1 at $M = \sqrt{2}$, but this close agreement is maintained at $M = \sqrt{1.25}$ only up to a reduced frequency of 0.05. Since both methods are based on the same linearized

equations of motion, this shows that the influence of frequency is greater at the lower supersonic Mach numbers.

Integrated values of lift and moment (L_3 and M_4) were also compared. Here differences as high as 3% were found at Mach numbers approaching one; however, for $M = \sqrt{2}$, the values were within 2%.

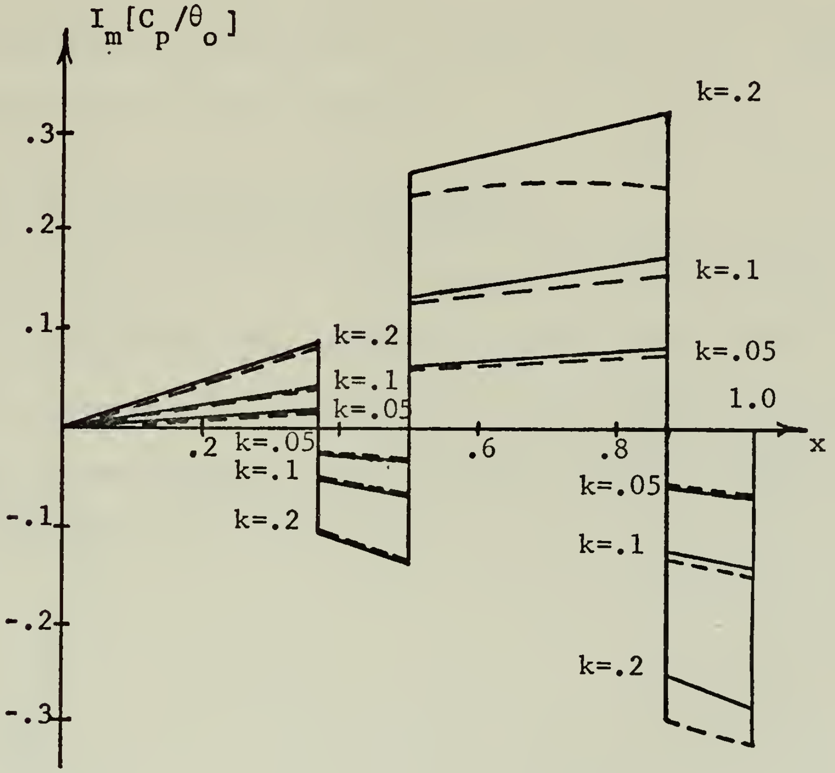


Figure 10

These results were produced using a grid fineness ratio of 100 or one quarter of the capacity of the program in order to reduce the amount of computer time required to make the computations.

F. COMPARISON OF METHOD OF CHARACTERISTICS WITH GARRICK-RUBINOW

Lift and moment calculations obtained from Programs A and B at $A \geq 1$ were compared with those of Garrick and Rubinow (1946) for both pitch and plunge oscillation. Differences were less than 2%. Program B was used to duplicate several different single-degree-of-freedom flutter points shown in Garrick and Rubinow's (1946) Figure 22. Again, the grid fineness ratio used was 100.

G. METHOD OF CHARACTERISTICS RESULTS

The following results were obtained using Program B to compute single-degree-of-freedom flutter speeds and frequencies. Approximately four minutes of CPU time on the IBM-360 were required to generate one flutter point. A grid fineness ratio of 100 was used throughout.

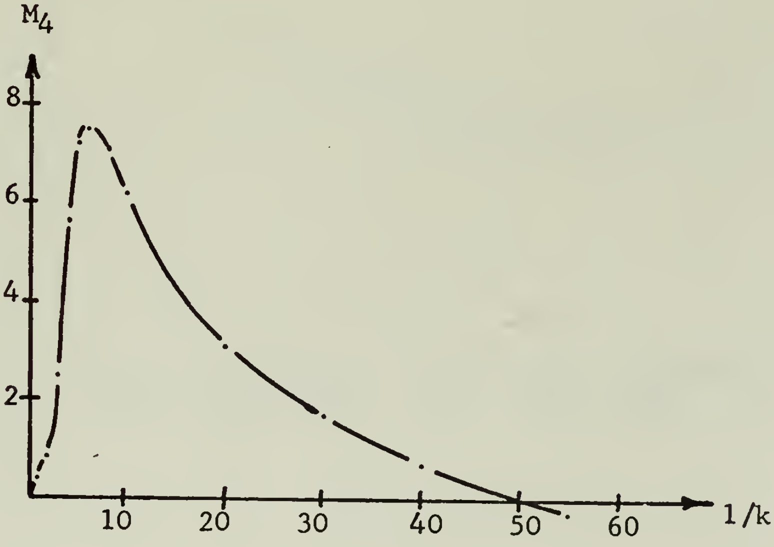


Figure 11

In Figure 11 ($M = 1.11$, $A = 0.95$, $\delta = 0$, $x_0 = .5$, $\mu = 15.708$, $r_\theta = .5$, $g = 0$) a typical variation of M_4 with $1/k$ is shown for an unstaggered cascade oscillating in phase with only a slight amount of blade interference. Here, the flutter point is at a relatively low value of reduced frequency ($k = 0.02$). That this is not the critical flutter condition is shown in Figure 12 where the non-dimensional flutter speed is plotted versus interblade phase angle for the same parameters as in Figure 11. In complete agreement with Lane (1956), the critical flutter condition, i.e., the lowest flutter speed, for this unstaggered cascade is at an interblade phase angle of 180°.

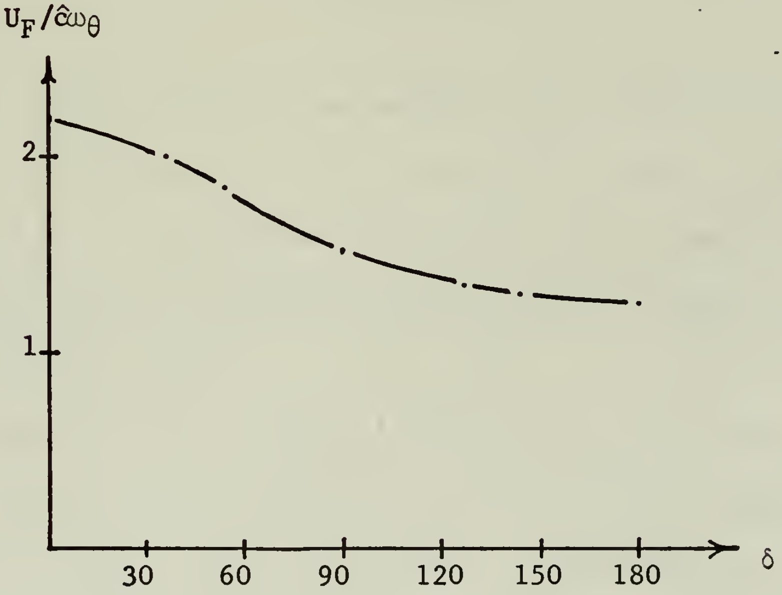


Figure 12

Figure 13 shows the effect of a small amount of interference on the flutter speed at M = 1.111 when plotted against elastic axis position.

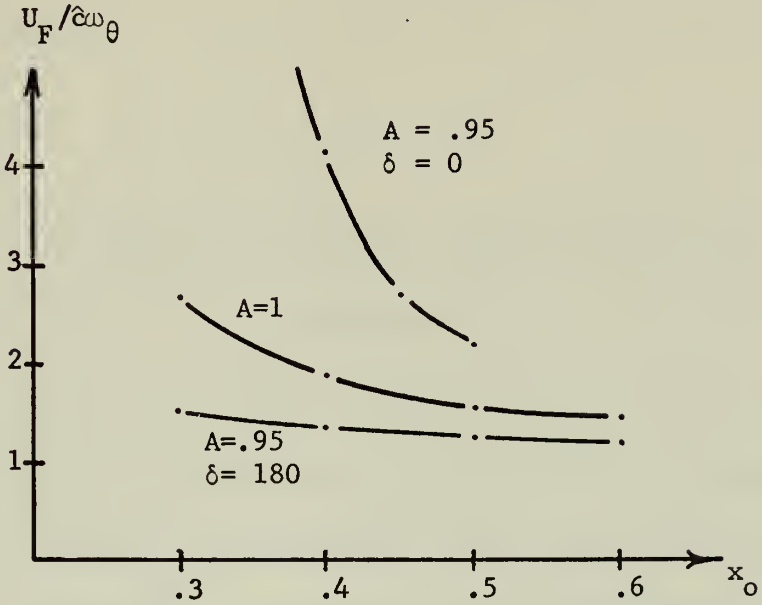


Figure 13

Again the blade and flow parameters are those as shown in Figure 11 and Figure 12. It can be seen that a relatively small amount of interference significantly lowers the non-dimensional flutter speed over the entire range of elastic axis positions tested.

In Figure 14 (A = .95, δ = 180, B = 0, μ = 500, g = 0, r_θ = .5) the effect of Mach number on flutter speed is shown for various elastic axis positions. An increase in Mach number to M = $\sqrt{2}$ significantly lowers the non-dimensional flutter speed for a relatively small amount of interference.

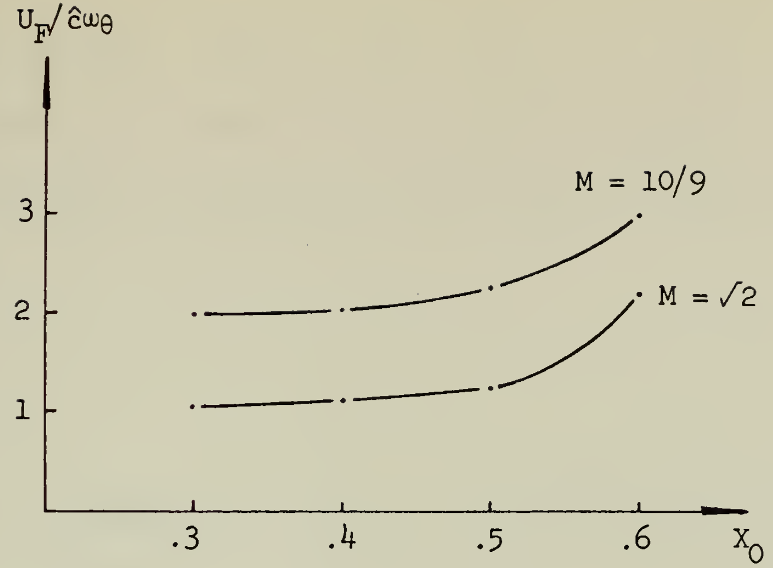


Figure 14

In Figure 15 (δ = 180, μ = 15.708, r_θ = .5, g = 0) the effect of this relatively small amount of interference on the flutter speed is again shown, but here at the higher Mach number (M = $\sqrt{2}$). As before, the non-dimensional flutter speed is significantly lowered over the entire elastic axis range by a small amount of interference.

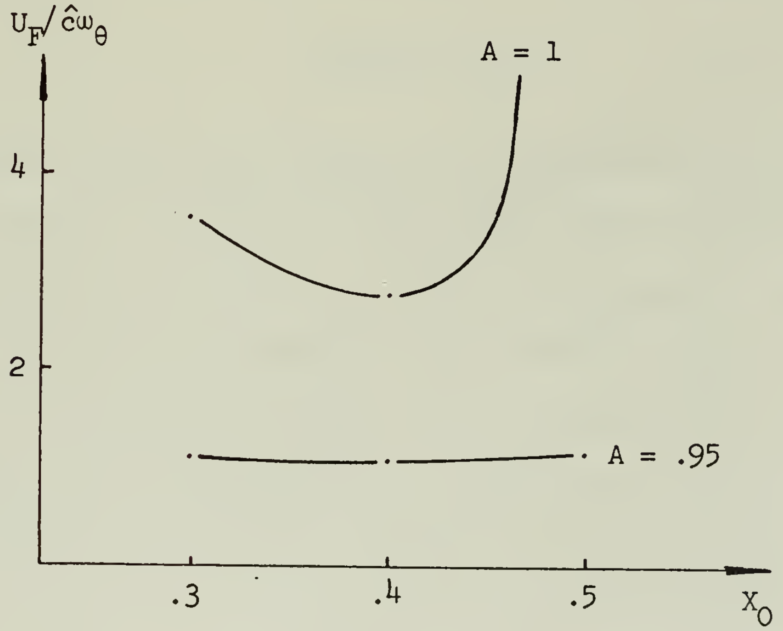


Figure 15

Figure 16 ($M = \sqrt{2}$, $A = .95$, $B = 0$, $\delta = 180$, $x_o = .5$, $r_\theta = .5$) shows the effect of structural damping on the non-dimensional flutter speed for two values of wing density parameter.

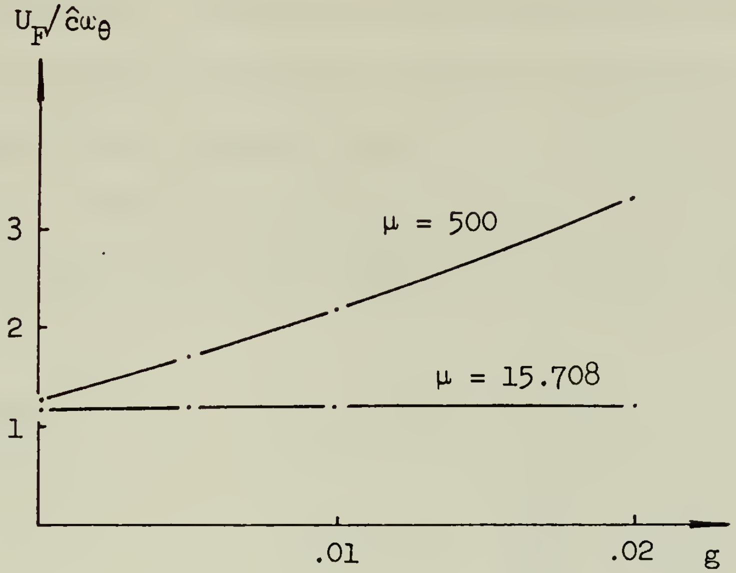


Figure 16

With a relatively small amount of interference, the flutter speed is greatly increased by a large wing density parameter and high structural damping. The high value of wing density parameter is more applicable to the cascade case. As defined in Garrick and Rubinow (1946), its increase may be interpreted as an increase in altitude for a fixed wing density, high values indicating supersonic wings at high altitude. For

typical supersonic cascades, the wing density parameter is of the order of 700.

The remaining results give variations in non-dimensional flutter speed of the staggered cascade (β = 26.565) at $M=\sqrt{2}$. Results shown are based on a cascade with an interblade phase angle of 180°. It must be re-emphasized that this interblade phase angle is not necessarily the one that gives the critical flutter condition; however, several points at random were chosen and the critical flutter condition found for each. All were within 1% of flutter speed at δ = 180 and within 10° of δ = 180. This was not done for all points shown in an effort to reduce the total amount of computer time used.

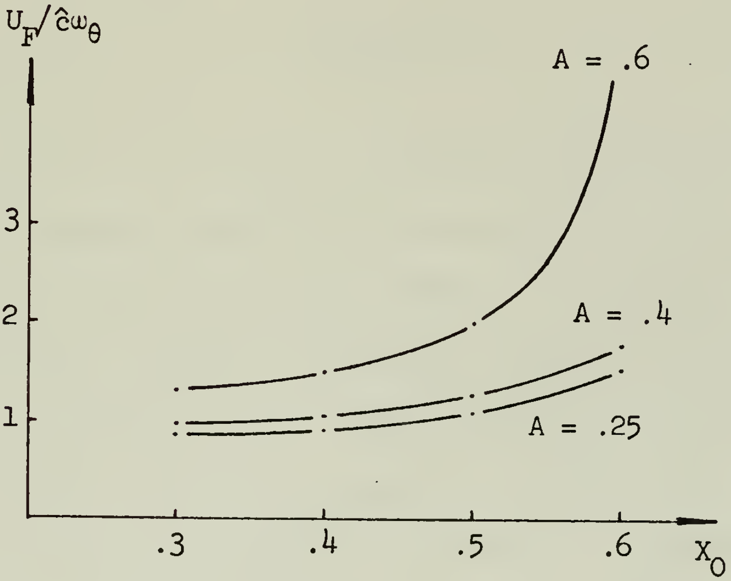


Figure 17

In Figure 17 (μ = 500, g = .02, r_θ = .5) the effect of an increasing amount of interference on flutter speed is shown for various elastic axis positions. The cascade is assumed

to have a high wing density parameter and 2% structural damping coefficient. An increase in the amount of interference is seen to lower the non-dimensional flutter speed.

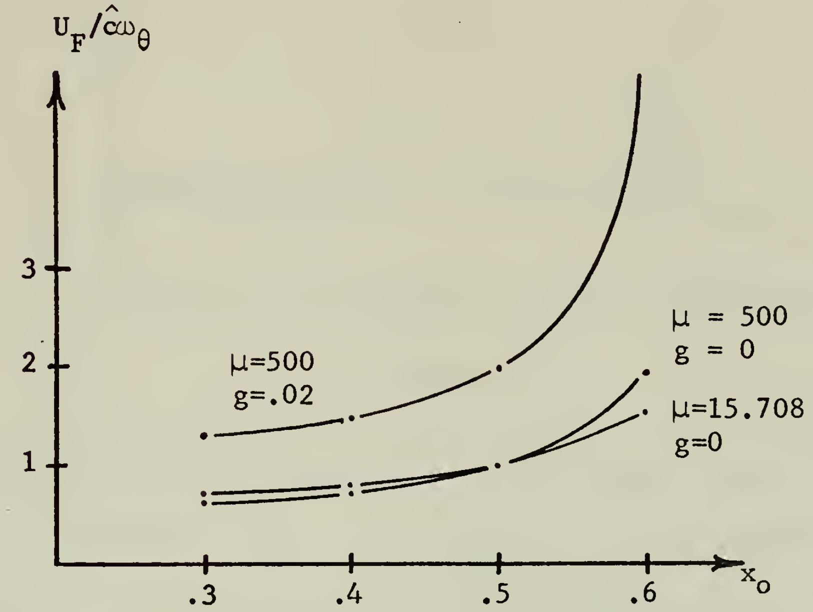


Figure 18

Figure 18 ($A = .6$, $r_\theta = .5$) shows the effect on the non-dimensional flutter speed of increasing the wing density parameter then the structural damping for various elastic axis positions. Without structural damping, an increase in wing density parameter had little effect on the flutter speed. With 2% structural damping, however, the flutter speed was greatly increased. The effect of structural damping is shown further in Figure 19 ($A = .6$, $x_0 = .5$, $r_\theta = .5$). Here, the variation of flutter speed with structural damping is shown for three values of wing density parameter. As can be seen, increasing the structural damping increases the

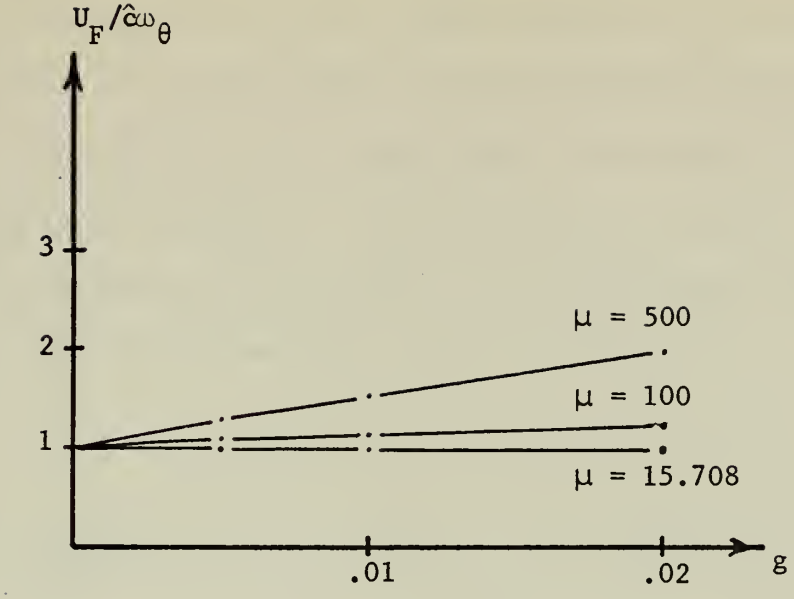


Figure 19

difference in flutter speeds for the three values of wing density parameter. Thus, the accurate determination of the wing density parameter becomes increasingly more important with increased structural damping.

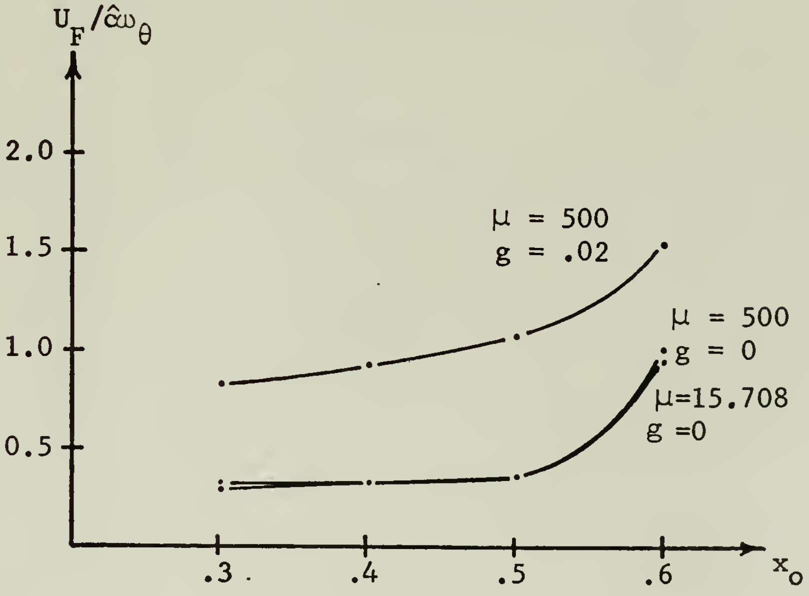


Figure 20

Figures 20 and 21 show the same parameters as Figures 18 and 19, respectively, but here, there is more interference (A = .25). The results are relatively the same as those shown in Figures 18 and 19. Flutter speeds overall are lower due to the increase in the amount of interference, and the effect of structural damping is more pronounced.

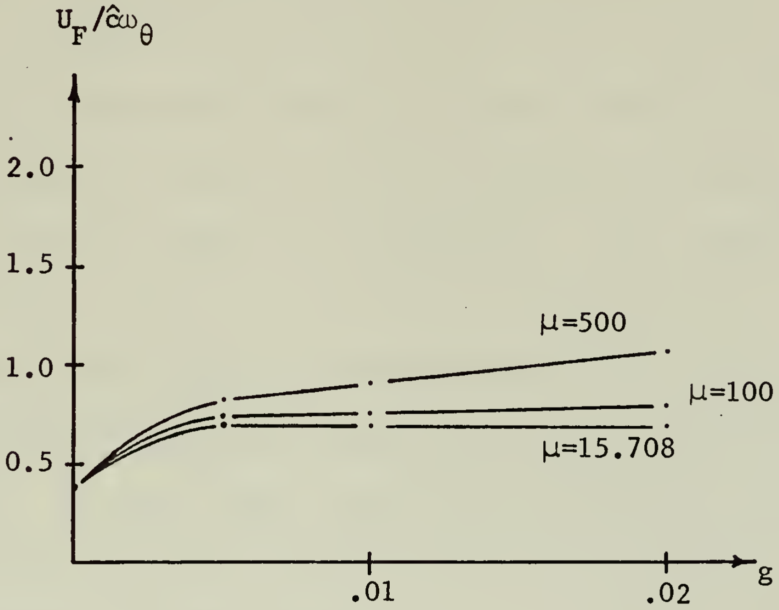


Figure 21

VIII. CONCLUSIONS AND RECOMMENDATIONS

The elementary theory for cascades with supersonic leading-edge locus is valid for low frequencies and can be used to give rapid calculations of lift forces and pitching moments of cascades as well as their torsional stability boundaries. These results can immediately be applied to a wide variety of wind tunnel and airfoil interference problems.

The method of characteristics can be used in conjunction with the high-speed computer to give rapid, accurate flutter calculations of cascades with supersonic leading-edge locus oscillating at arbitrary frequency. This applies to torsion and bending flutter calculations as well as to torsion flutter alone, for the only differences in the calculations are those due to different values of normal velocity at the blade surface. In each case, torsion or bending, the method of calculating the flow field and determining the lift forces and moments is identical.

Results of single-degree-of-freedom (torsional) flutter calculations show that interference on cascade blades due to reflected Mach waves from adjacent blades lowers flutter speeds significantly and in proportion to the amount of this interference. The presence of structural damping, however, can partially alleviate this adverse situation by increasing flutter speeds, provided the wing density parameter is large.

Recommendations for future study include the development of the elementary theory for bending oscillations, and the generation of numerical results for two-degree-of-freedom flutter using Program A. Of primary importance, however, is the development of a method of characteristics approach for cascades with subsonic leading-edge locus. This is the critical problem today in the design of the supersonic compressor. With this problem properly formulated, the method of characteristics should be able to give rapid, accurate flutter predictions for cascades oscillating in this complicated flow field.

APPENDIX A

COMPUTER PROGRAM VARIABLES

I. LOGIC VARIABLES

The following variables and their definitions, in alphabetical order appear in the logic statements of the programs. The asterisk is placed behind those variables that appear only in Program A.

FCONST*	Denotes type of oscillation: TRUE - plunge mode, FALSE - pitch mode.
IATNWL	When computation reaches upper airfoil, it takes on the value "1", otherwise, "0".
ICO	Takes on the value "1" when lower airfoil computation is complete, otherwise "0".
ICOUNT	Incremented only when MACHLN is called by being set equal to JLINE, zeroed when HIFOIL is called.
IFIN	Set equal to "1" when integration on lower airfoil is complete, otherwise "0".
IHAVEP	The number of the computation in any row starting with "0".
IJUNC	The grid point at the intersection of the initial right and left-running Mach lines.
IJUNT	Equal to IJUNC, needed to preserve the value initially calculated in case of multiple flow field calculations.
ILINE	The number of grid points in the row (right-running Mach line) being computed, plus one.
IMAGRT*	If X is negative, set equal to "1", otherwise "0".
IREF	Set equal to "1" if lower airfoil point is in Zone I, otherwise "0".
ISPLIT	Set equal to 1/2 of the number of grid points minus the number of stagger points.

ISWICH	Set equal to "1" or "2" alternately starting with "1" and changing when upper airfoil calculation goes into a different zone. The upper airfoil zone marker.
ISWTCH	Switch variable used to choose the row being calculated, alternately set equal to "1" or "2". Used as a second subscript.
ITEM	Loop variable for pitch and plunge mode calculation.
IZOT*	Set equal to "1" if flutter frequency ratio is imaginary, otherwise "0".
JCOUNT	Set equal to "0" when in Zone I, incremented when MACHLN is called in an interference region.
JFIN	Set equal to "1" when integration along upper airfoil is complete, otherwise "0".
JLINE	The number of grid points in the row being computed.
JSWICH	The same as ISWICH but used in LOFOIL. The lower airfoil zone marker.
JSWTCH	Switch variable used to choose the preceding line that was calculated, set equal to the opposite value as that of ISWTCH. Used as a second subscript.
JTEM	Loop variable for frequency range incrementing.
KCOUNT	Used in the interference calculation in MACHLN, set equal to "JCOUNT-1".
KOUNT	Set equal to "0" until HIFOIL is called, then set equal to the number of grid points minus one.
LCOUNT	The number starting with "0" of each point in a zone of interference in LOFOIL.
MCOUNT	Same as LCOUNT but used in HIFOIL.

II. QUANTITY VARIABLES

The following variables in the programs take on the values as defined below. All have real values unless otherwise stated. The asterisk is placed behind those variables that appear only in Program A and the accent is placed behind those variables that appear only in Program B. The dimensioned variables are listed first.

U22R, U22I, V22R, V22I, C22R, C22I(400,3).	The velocities, real and imaginary, of a point in the program. They are equivalent to the respective Teipel amplitude functions, U_{22R} through C_{22I}.
U33R, U33I, V33R, V33I, C33R, C33I(200,2).	Same as above, but used to calculate the flow field in Zone I' when the phase angle is non-zero.
X, Y(400,2)	The x and y coordinates of a point based on the chord length.
AI	A_I as defined in Eq. (96a).
BI	B_I as defined in Eq. (96b).
CAPX'	X as defined in the Table of Symbols.
CPI1	The imaginary part of $e^{-i\delta}(C_{22R}+iC_{22I})$ of the preceding point calculated on the upper airfoil.
CPR1	The real part of $e^{-i\delta}(C_{22R}+iC_{22I})$ of the preceding point calculated on the upper airfoil.
CPI2	The imaginary part of $e^{-i\delta}(C_{22R}+iC_{22I})$ of the point being calculated on the upper airfoil.
CPR2	The real part of $e^{-i\delta}(C_{22R}+iC_{22I})$ of the point being calculated on the upper airfoil.
DAMP'	The total system damping as defined in Eq. (218b).
DELCI	The value of C_{22I} at $x = 1$.

DELCPI	The imaginary part of $e^{-i\delta}$(DELCR +iDELCI) on the upper airfoil.
DELCPR	The real part of $e^{-i\delta}$(DELCR + iDELCI) on the upper airfoil.
DELCR	The value of C_{22R} at $x = 1$.
DELL1, DELL2, DELM1, DELM2.	The change in lift (real and imaginary) and moment, respectively, due to extrapolation to $x = 1$.
DELTA	The phase angle in radians.
DELTAS	The size of the length increment along a Mach line.
DISC*	The discriminant of Eq. (206).
DSTSTR	The size of the Δx increment.
FACTOR'	Set equal to $\Delta x/k^2$.
FACTOR*	In the MAIN, set equal to $\Delta x/k^2$ for the pitch mode, and set equal to $1/2\ \Delta x/k^2$ for the plunge mode. In FLUTER, set equal to DISC - C_R.
FAZE	The phase angle in degrees.
FIN	The chord length.
FMANGL	The Mach angle in radians.
FNGDPT	The real number value of the number of grid points on the initial Mach line.
FNGRDN	The real number value of the grid fineness ratio, equal to FNGDPT.
FSTRMN	The freestream Mach number.
G1,...,G9	Intermediate factors used in velocity calculations in MACHLN, LOFOIL, GENFPT, and HIFOIL.
GSUBA*	The structural damping coefficient, g, as defined in the Table of Symbols.
HAFREQ*	The ratio of the natural frequency of the airfoil in plunge to that in pitch.
HDSTRL	The increment, $\Delta x/2$.

INCRE	The amount that 2/k is incremented each time the lift and moment is calculated.
K12R,...,K56I	$K_{12R},\ldots,K_{56I}$ as defined in Eq. (95).
L1, L2	The real and imaginary lift coefficients, respectively. They become L_1 and L_2 or L_3 and L_4 as defined in Eq. (186) and Eq. (187) depending on the type of oscillation.
L3, L4*	The real and imaginary lift coefficients, respectively, due to pitching oscillations.
M1, M2	The real and imaginary moment coefficients, respectively. They become M_1 and M_2 or M_3 and M_4 as defined in Eq. (188) and Eq. (189) depending on the type of oscillation.
M3, M4*	The real and imaginary moment coefficients, respectively, due to pitching oscillations.
MUU	The wing density parameter as defined in the Table of Symbols.
NGDPTS	The grid fineness ratio plus one.
NGRDFN	The grid fineness ratio, it must be an even number less than 400. It is an integer.
NSTPTS	The number of stagger grid points, an integer, set equal to $B/\Delta x$ plus one.
OMEGA	The non-dimensional flutter frequency as defined in Eq. (208).
OMEGAA	Ω_θ as defined in the Table of Symbols.
OMEGAH*	Ω_h as defined in the Table of Symbols.
REDFRQ	The reduced frequency.
RFREQ	2/k.
ROOTX	The square root of X.
ROOTX1*	The square root of the first root of Eq. (206).
ROOTX2*	The square root of the second root of Eq. (206).
ROOTXI*	The square root of Eq. (207).

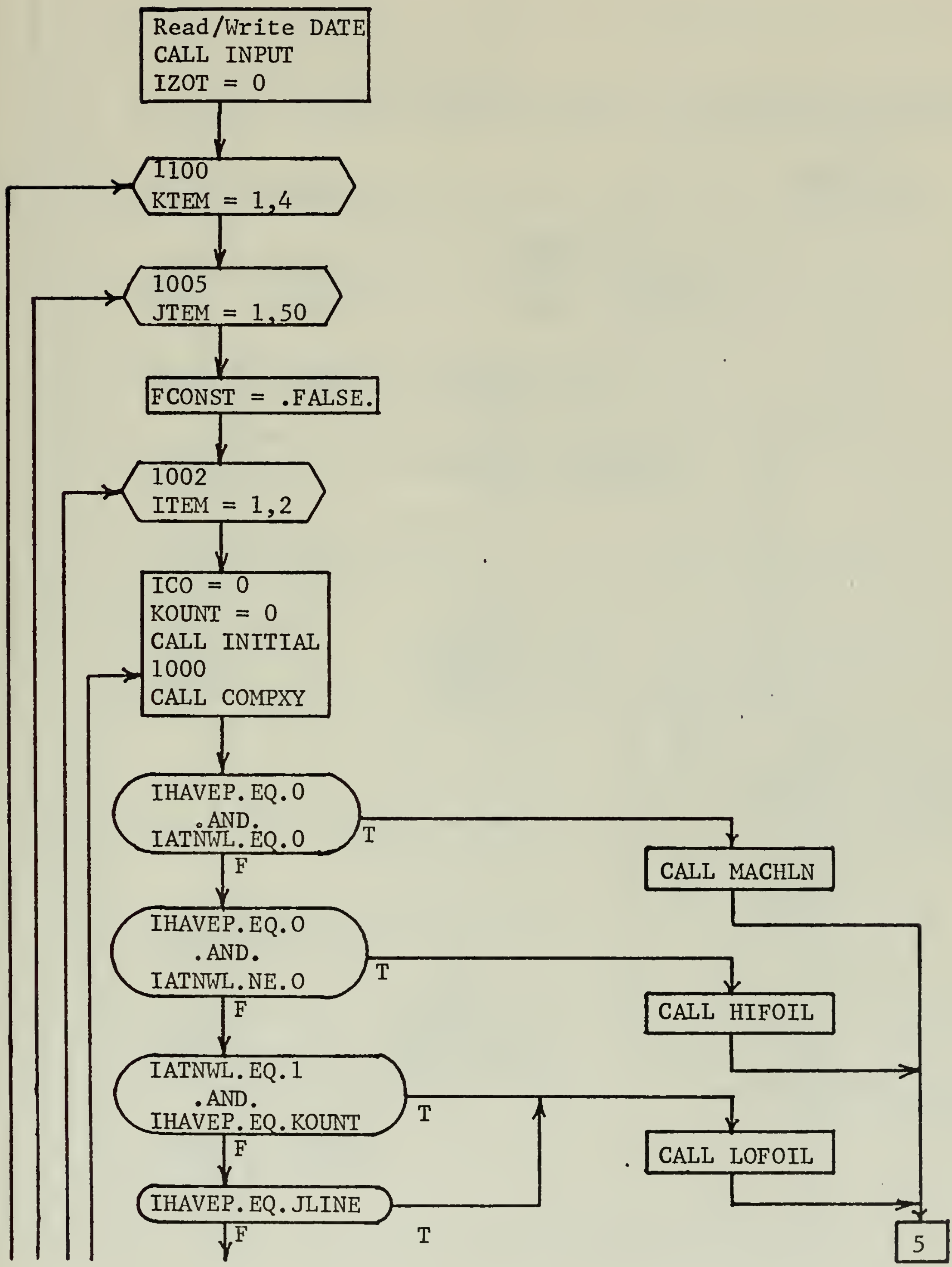
RSUBA	The radius of gyration, r_θ, as defined in the Table of Symbols.
RTOSPH	The ratio of specific heats.
S	The square root of M^2-1.
STGANG	The stagger angle in degrees.
STGR	The amount of stagger, B.
T	M^2-1.
TEST1, TEST2*	The difference between the respective roots of Eq. (206) and that of Eq. (207).
TEST3, TEST4*	The values of TEST1 and TEST2, respectively, for the previous frequency computation.
TNWDST	The perpendicular distance between the two airfoils, d.
TOPCRD	The chord length of the top airfoil, x-B.
TRNGLH	The increment, $\frac{1}{2}\Delta x \cos \alpha$.
U	The factor, $\cos(k \frac{M^2}{M^2-1} x)$.
UF	The non-dimensional flutter speed as defined in Eq. (209).
UR, UI, VR, VI, CR, CI	The values of $U_{11R},\ldots,C_{11I}$, respectively, for calculations on the initial Mach line between the point of intersection of the initial left and right-running Mach lines and the upper airfoil.
V	The factor, $\sin (k \frac{M^2}{M^2-1} x)$.
VIPANL	The imaginary part of the normal flow velocity at the airfoil surface.
VRPANL	The real part of the normal flow velocity at the airfoil surface.
W	M^2.
XI*	X_I as defined in Eq. (207).
XLENGTH	The length of the initial Mach line.
XR1, XR2*	The roots of Eq. (206).

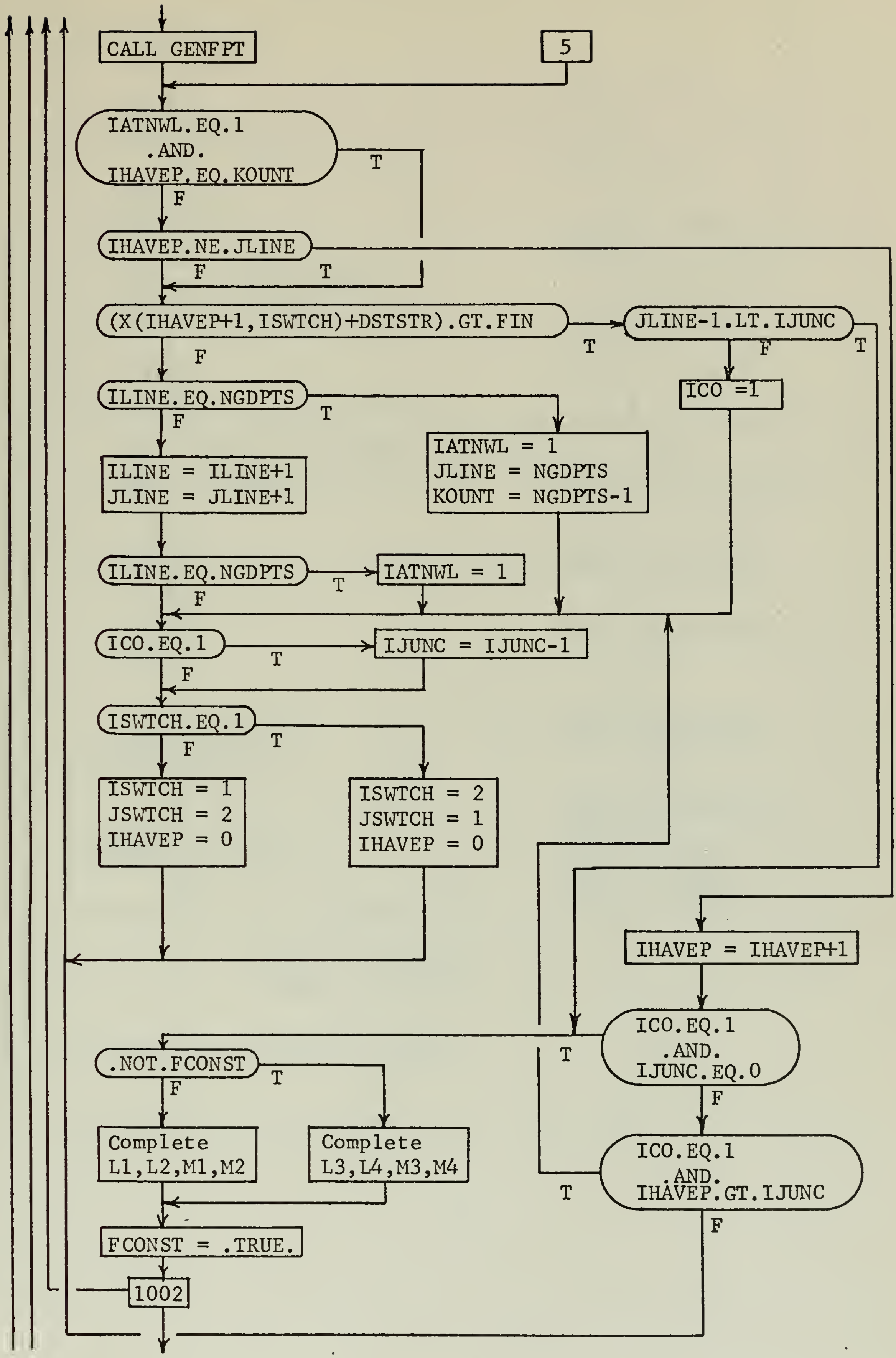
XSUBA*	The distance from the elastic axis of the airfoil to its center of gravity.
XSUBO	The distance from the leading-edge of the airfoil to its elastic axis.

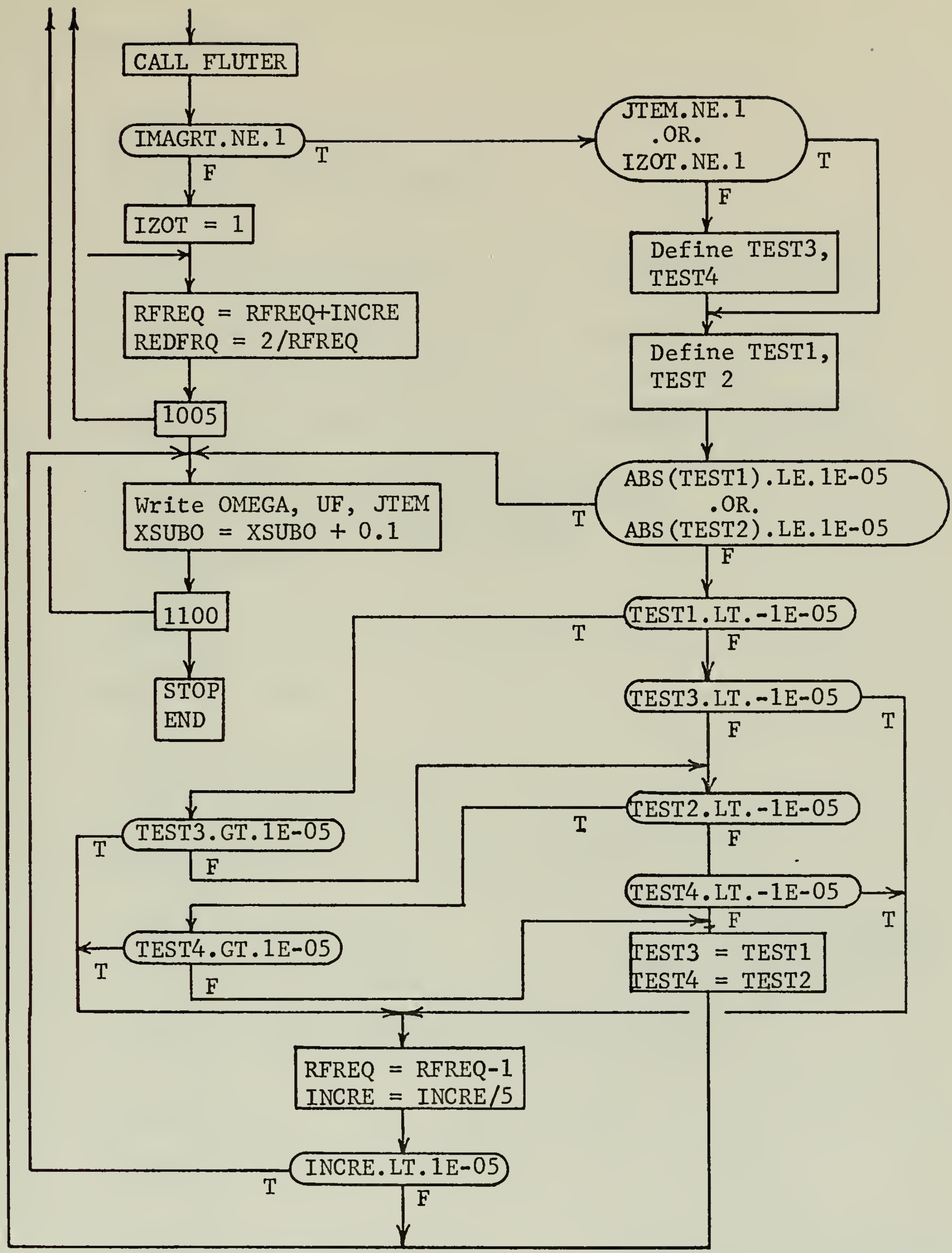
APPENDIX B

FLOW DIAGRAM - PROGRAM A

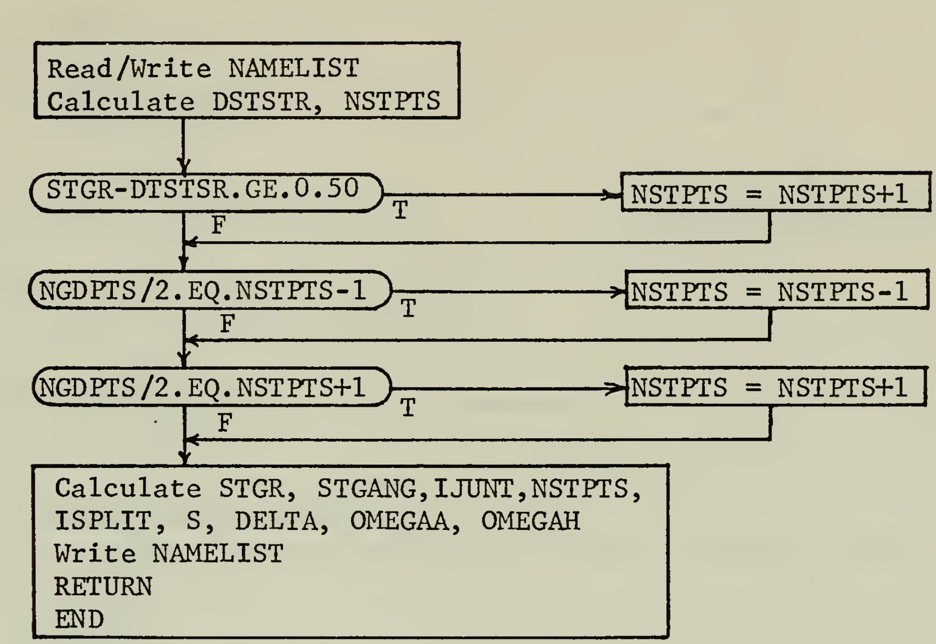
I. MAIN PROGRAM



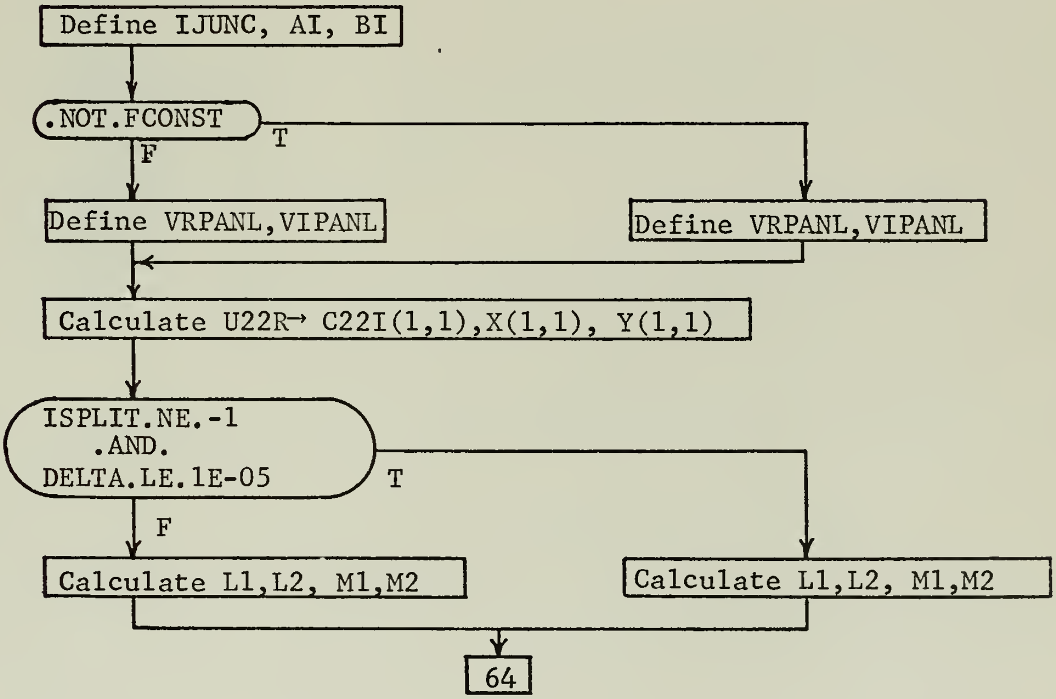

CALL GENFPT
5
IATNWL.EQ.1 .AND. IHAVEP.EQ.KOUNT
T
F
IHAVEP.NE.JLINE
F
T
(X(IHAVEP+1,ISWTCH)+DSTSTR).GT.FIN
T
JLINE-1.LT.IJUNC
F
T
F
ICO =1
ILINE.EQ.NGDPTS
F
T
IATNWL = 1
JLINE = NGDPTS
KOUNT = NGDPTS-1
ILINE = ILINE+1
JLINE = JLINE+1
ILINE.EQ.NGDPTS
T
IATNWL = 1
F
ICO.EQ.1
T
IJUNC = IJUNC-1
F
ISWTCH.EQ.1
F
T
ISWTCH = 1
JSWTCH = 2
IHAVEP = 0
ISWTCH = 2
JSWTCH = 1
IHAVEP = 0
IHAVEP = IHAVEP+1
ICO.EQ.1 .AND. IJUNC.EQ.0
T
F
.NOT.FCONST
F
T
Complete L1,L2,M1,M2
Complete L3,L4,M3,M4
ICO.EQ.1 .AND. IHAVEP.GT.IJUNC
T
F
FCONST = .TRUE.
1002

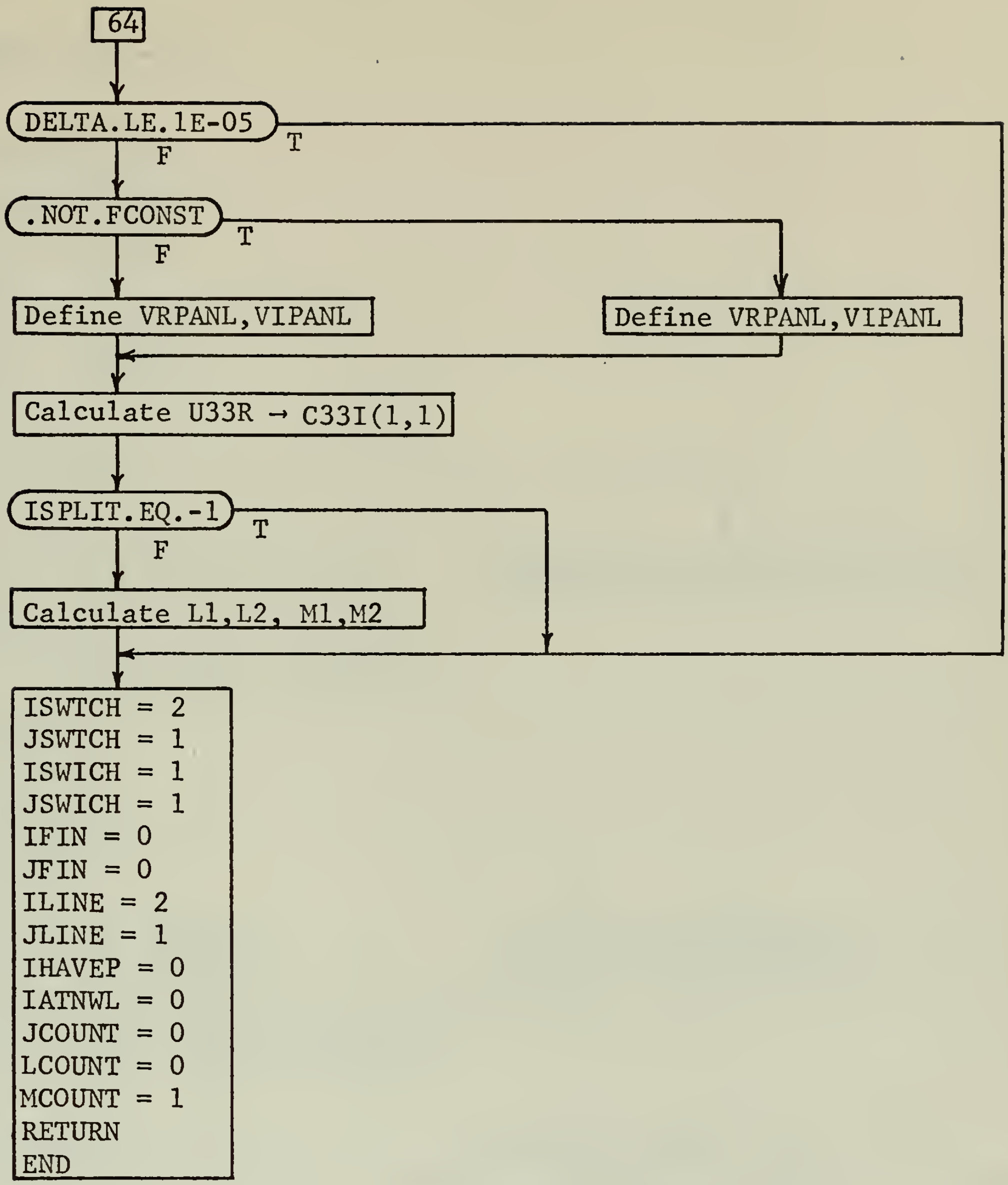

CALL FLUTER
IMAGRT.NE.1
T
F
IZOT = 1
RFREQ = RFREQ+INCRE
REDFRQ = 2/RFREQ
1005
Write OMEGA, UF, JTEM
XSUBO = XSUBO + 0.1
1100
STOP
END
JTEM.NE.1
.OR.
IZOT.NE.1
T
F
Define TEST3,
TEST4
Define TEST1,
TEST 2
ABS(TEST1).LE.1E-05
.OR.
ABS(TEST2).LE.1E-05
T
F
TEST1.LT.-1E-05
T
F
TEST3.LT.-1E-05
F
T
TEST3.GT.1E-05
T
F
TEST2.LT.-1E-05
T
F
TEST4.LT.-1E-05
F
T
TEST4.GT.1E-05
T
F
TEST3 = TEST1
TEST4 = TEST2
RFREQ = RFREQ-1
INCRE = INCRE/5
INCRE.LT.1E-05
T
F

II. SUBROUTINE INPUT

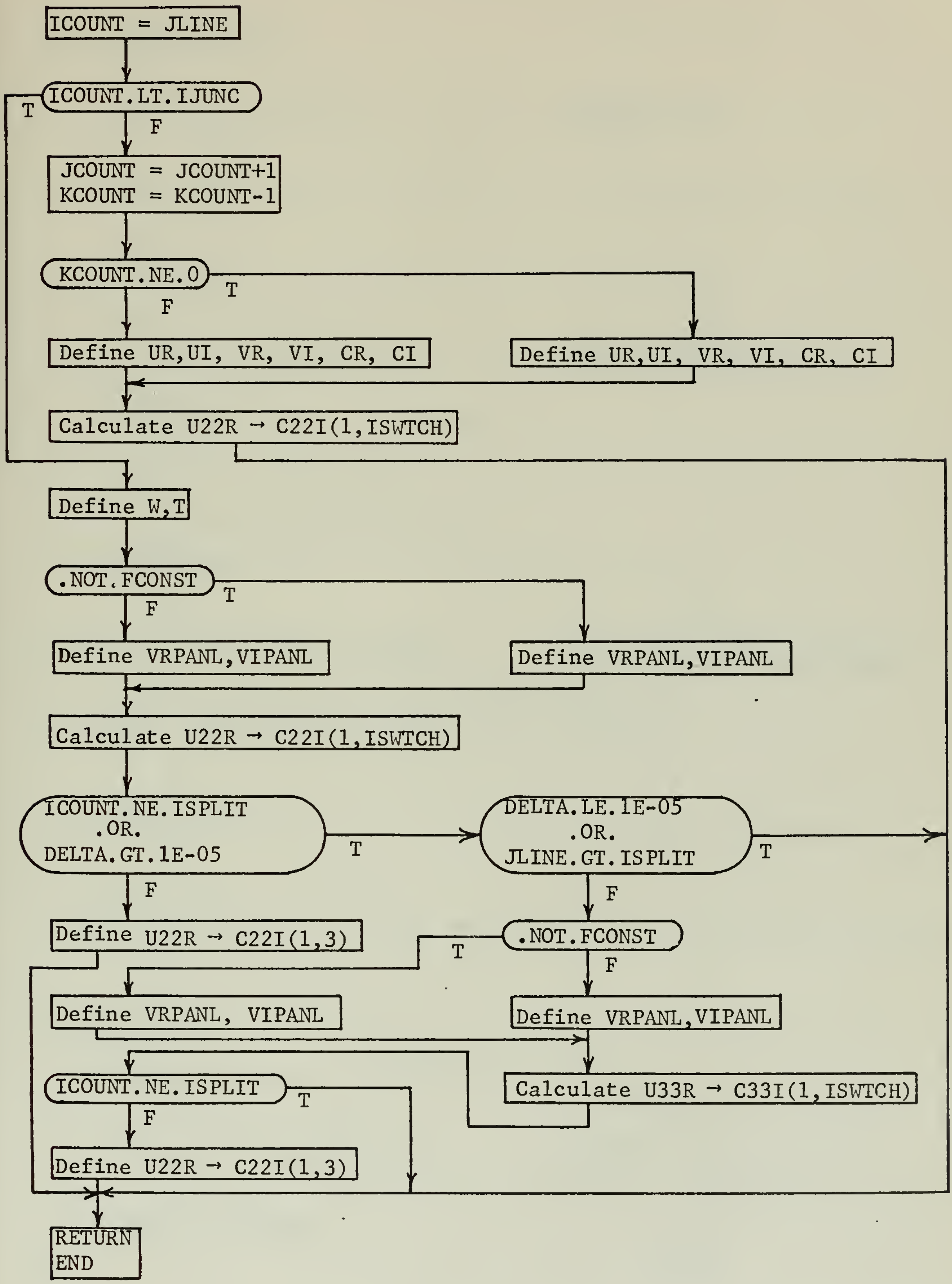


III. SUBROUTINE INTIAL

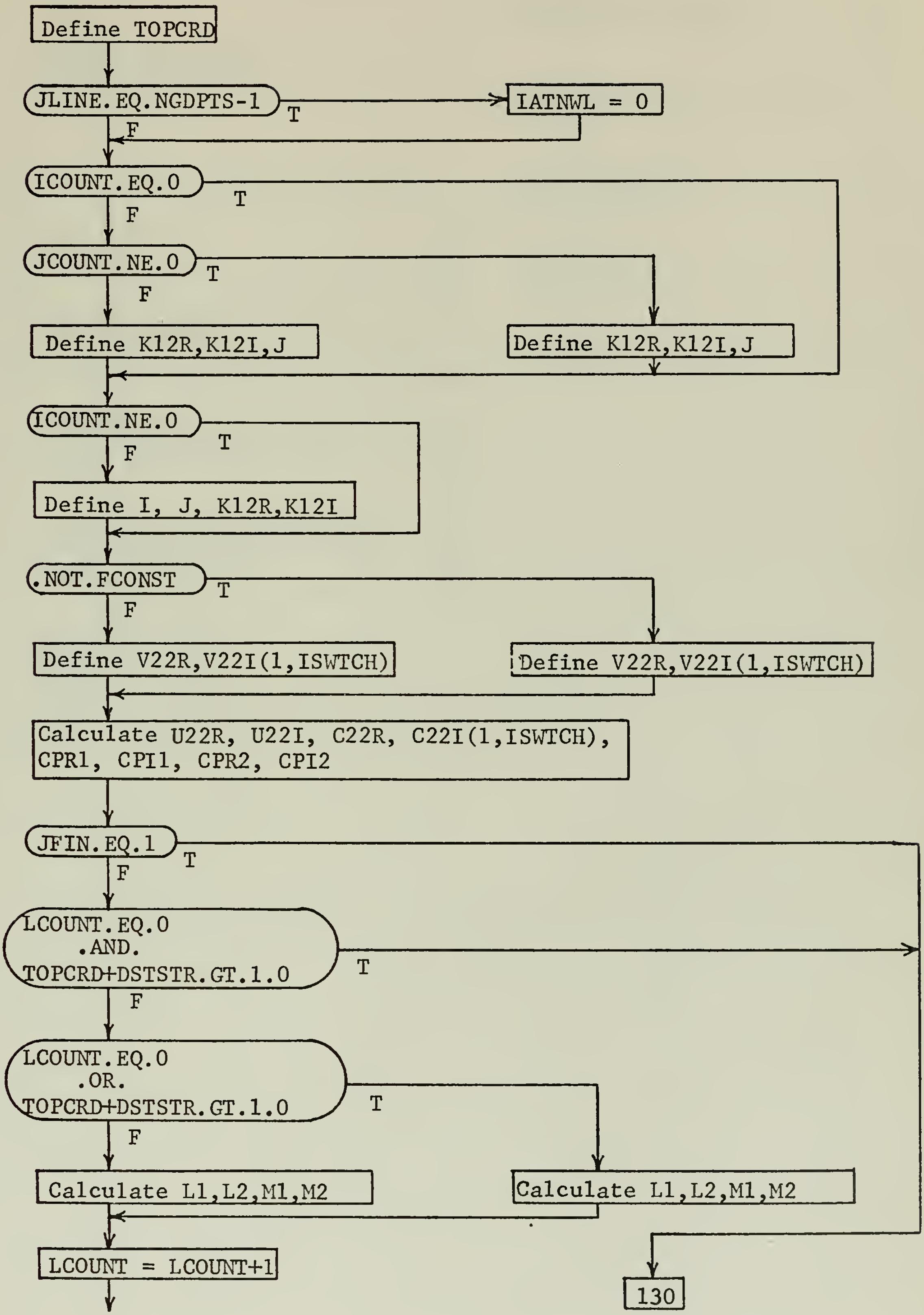


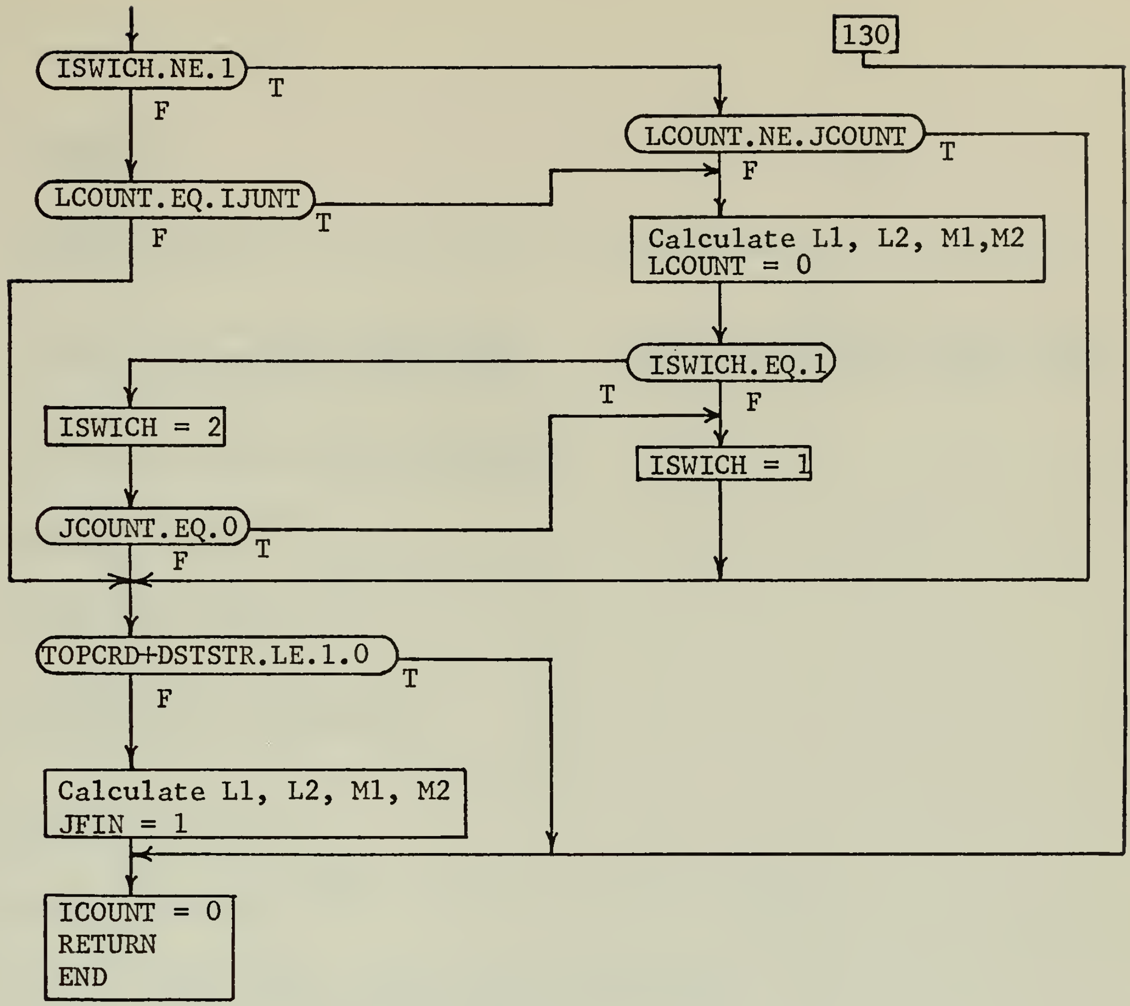

64
DELTA.LE.1E-05
T
F
.NOT.FCONST
T
F
Define VRPANL,VIPANL
Define VRPANL,VIPANL
Calculate U33R → C33I(1,1)
ISPLIT.EQ.-1
T
F
Calculate L1,L2, M1,M2
ISWTCH = 2
JSWTCH = 1
ISWICH = 1
JSWICH = 1
IFIN = 0
JFIN = 0
ILINE = 2
JLINE = 1
IHAVEP = 0
IATNWL = 0
JCOUNT = 0
LCOUNT = 0
MCOUNT = 1
RETURN
END

IV. SUBROUTINE MACHLN



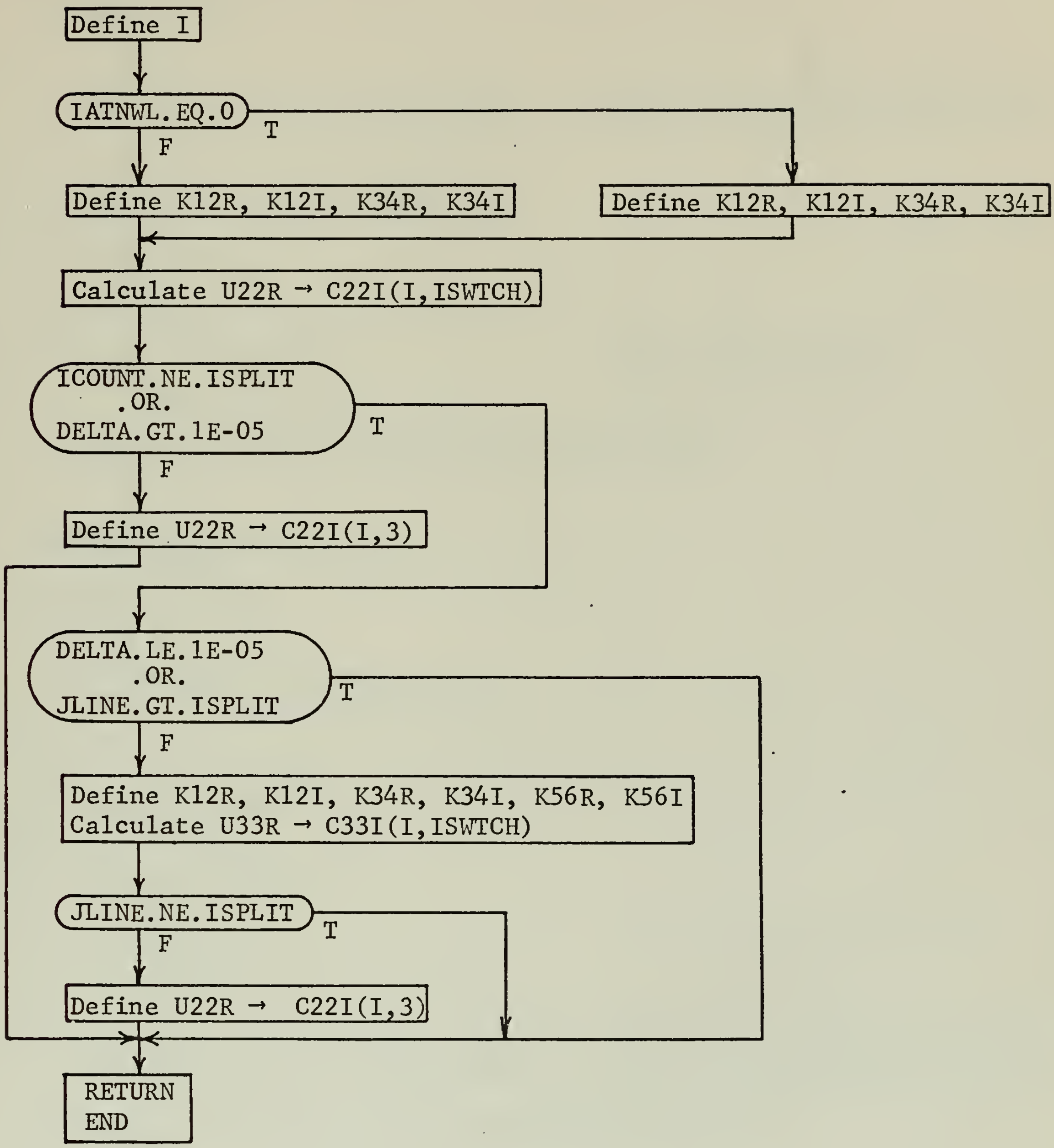
V. SUBROUTINE HIFOIL



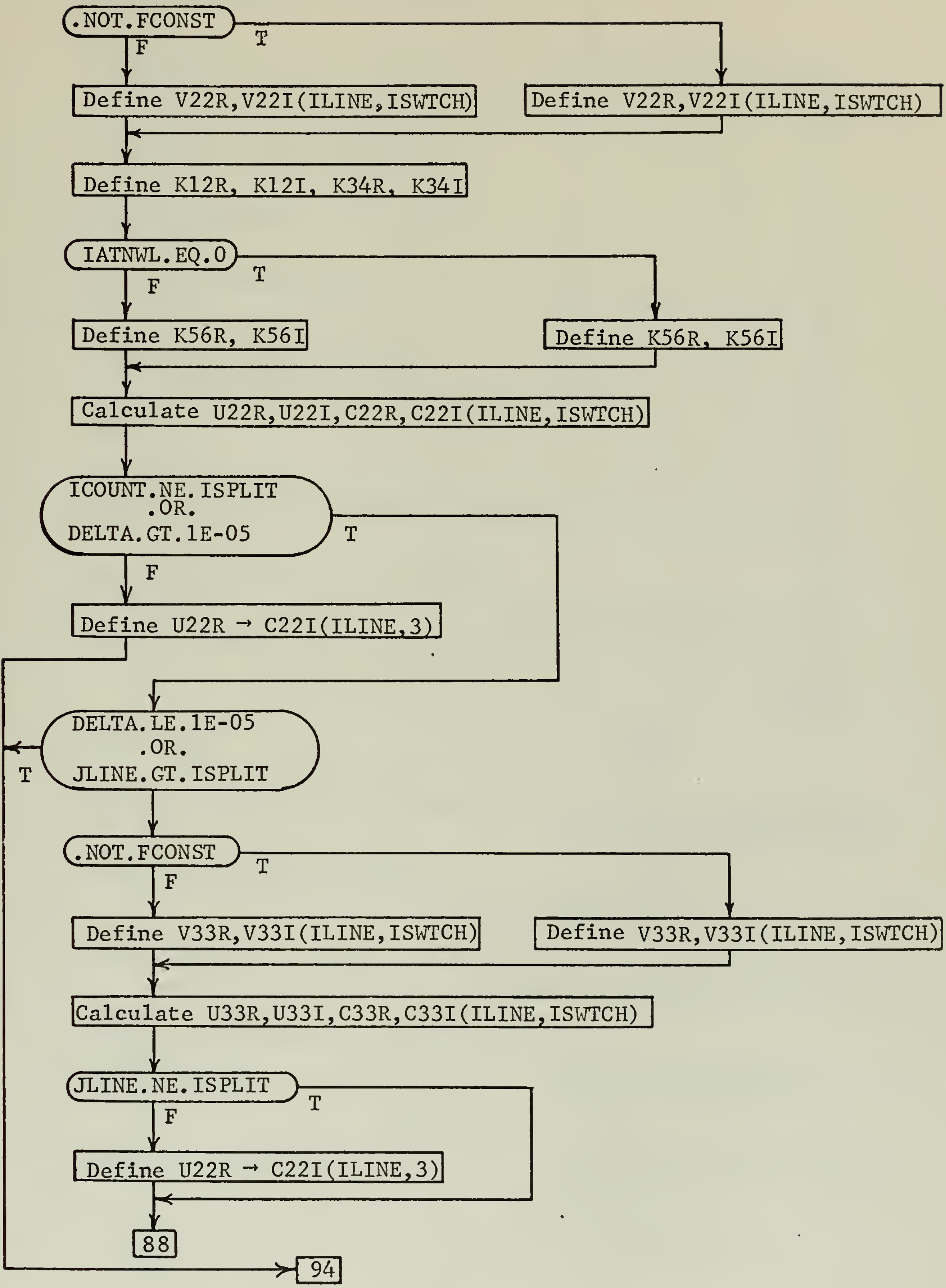


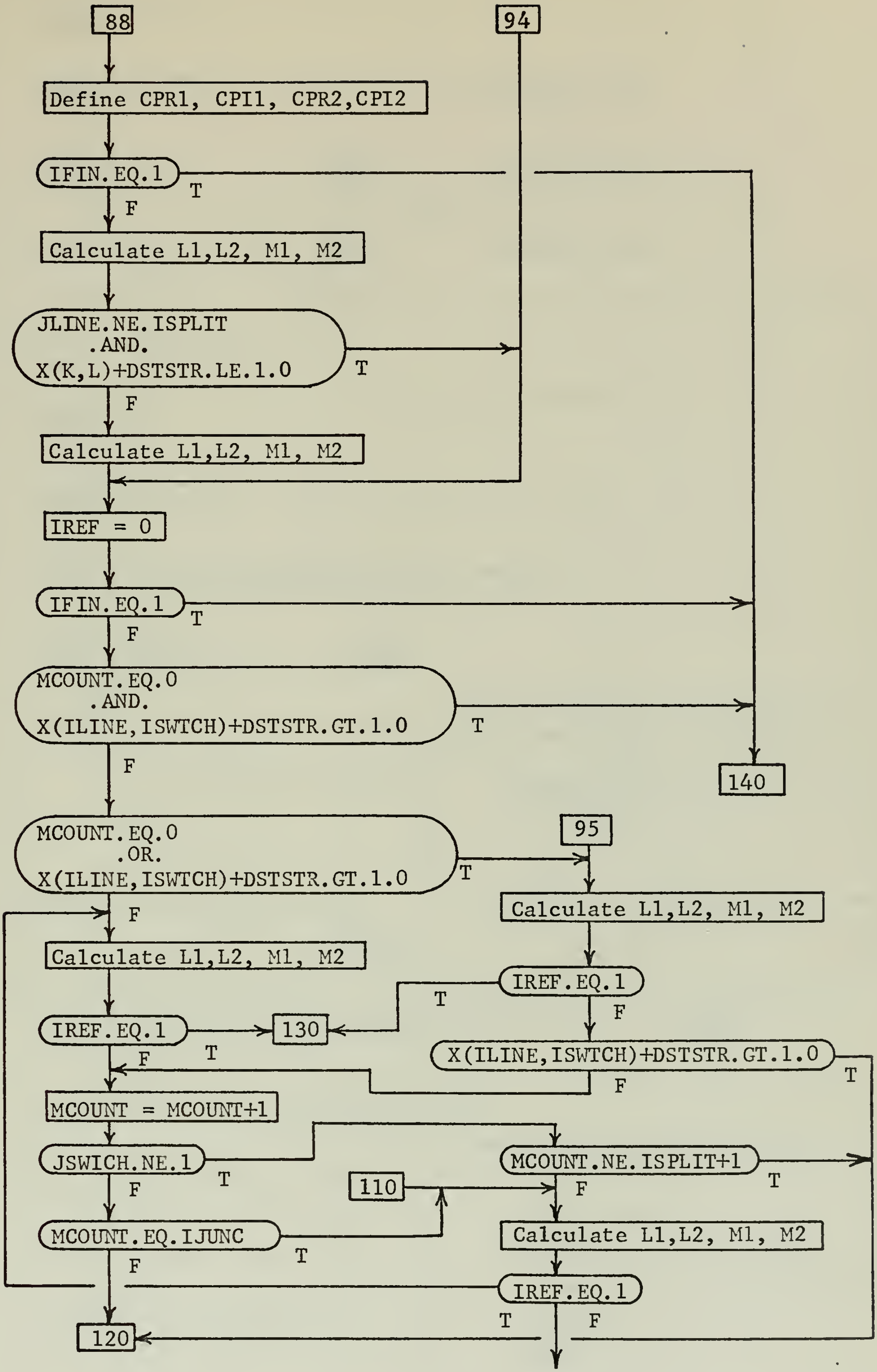
130
ISWICH.NE.1
T
F
LCOUNT.NE.JCOUNT
T
F
LCOUNT.EQ.IJUNT
T
F
Calculate L1, L2, M1,M2
LCOUNT = 0
ISWICH.EQ.1
T
F
ISWICH = 2
ISWICH = 1
JCOUNT.EQ.0
T
F
TOPCRD+DSTSTR.LE.1.0
T
F
Calculate L1, L2, M1, M2
JFIN = 1
ICOUNT = 0
RETURN
END

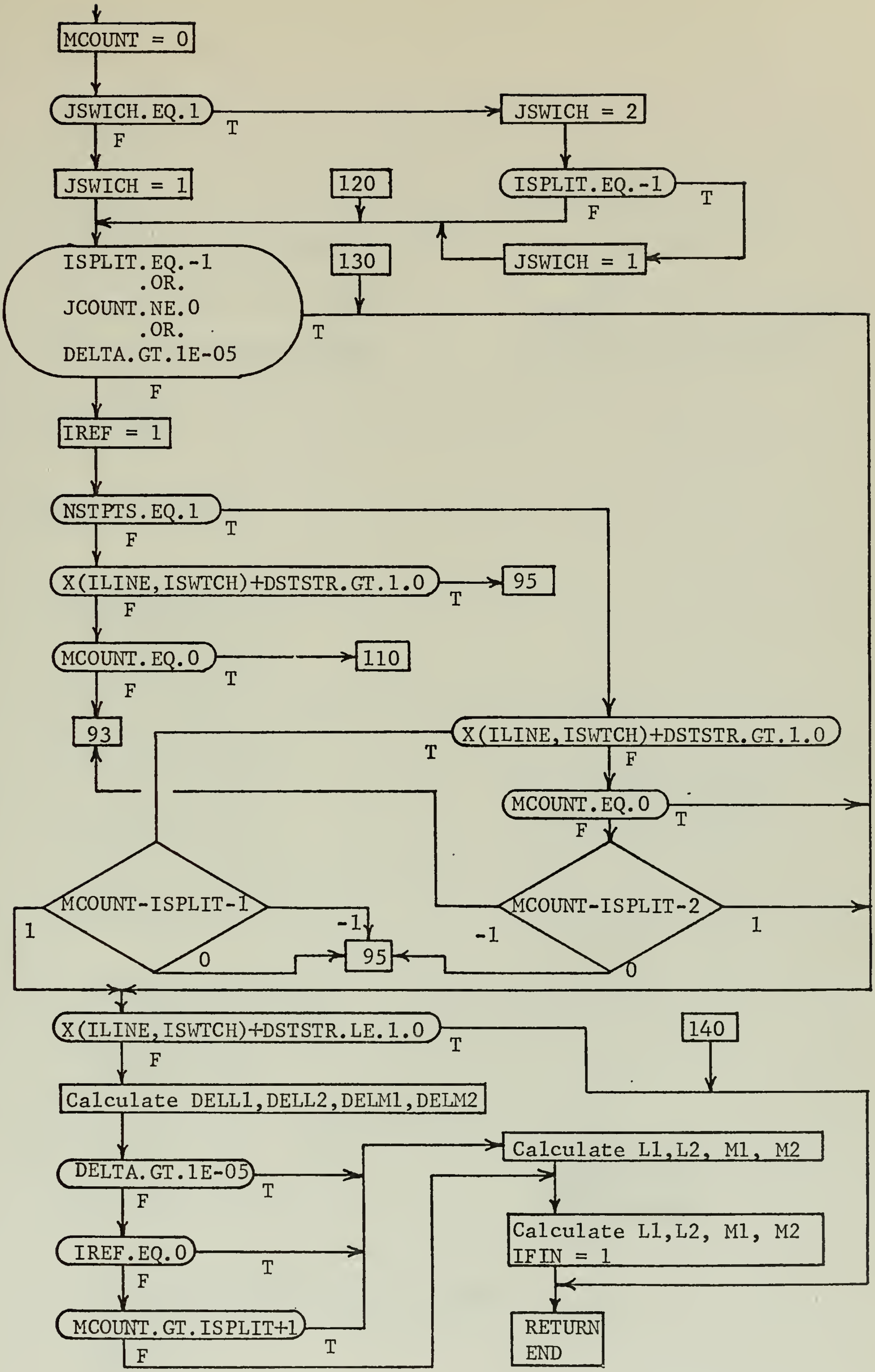
VI. SUBROUTINE GENFPT


Define I
IATNWL.EQ.0
T
F
Define K12R, K12I, K34R, K34I
Define K12R, K12I, K34R, K34I
Calculate U22R → C22I(I,ISWTCH)
ICOUNT.NE.ISPLIT .OR. DELTA.GT.1E-05
T
F
Define U22R → C22I(I,3)
DELTA.LE.1E-05 .OR. JLINE.GT.ISPLIT
T
F
Define K12R, K12I, K34R, K34I, K56R, K56I
Calculate U33R → C33I(I,ISWTCH)
JLINE.NE.ISPLIT
T
F
Define U22R → C22I(I,3)
RETURN
END

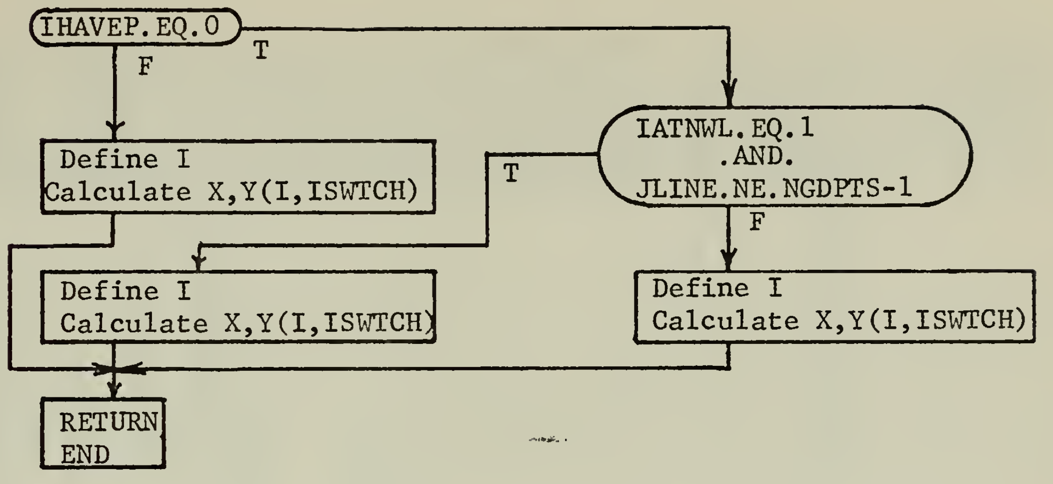
VII. SUBROUTINE LOFOIL


.NOT.FCONST
F
T
Define V22R,V22I(ILINE,ISWTCH)
Define V22R,V22I(ILINE,ISWTCH)
Define K12R, K12I, K34R, K34I
IATNWL.EQ.0
F
T
Define K56R, K56I
Define K56R, K56I
Calculate U22R,U22I,C22R,C22I(ILINE,ISWTCH)
ICOUNT.NE.ISPLIT .OR. DELTA.GT.1E-05
T
F
Define U22R → C22I(ILINE,3)
DELTA.LE.1E-05 .OR. JLINE.GT.ISPLIT
T
.NOT.FCONST
F
T
Define V33R,V33I(ILINE,ISWTCH)
Define V33R,V33I(ILINE,ISWTCH)
Calculate U33R,U33I,C33R,C33I(ILINE,ISWTCH)
JLINE.NE.ISPLIT
F
T
Define U22R → C22I(ILINE,3)
88
94

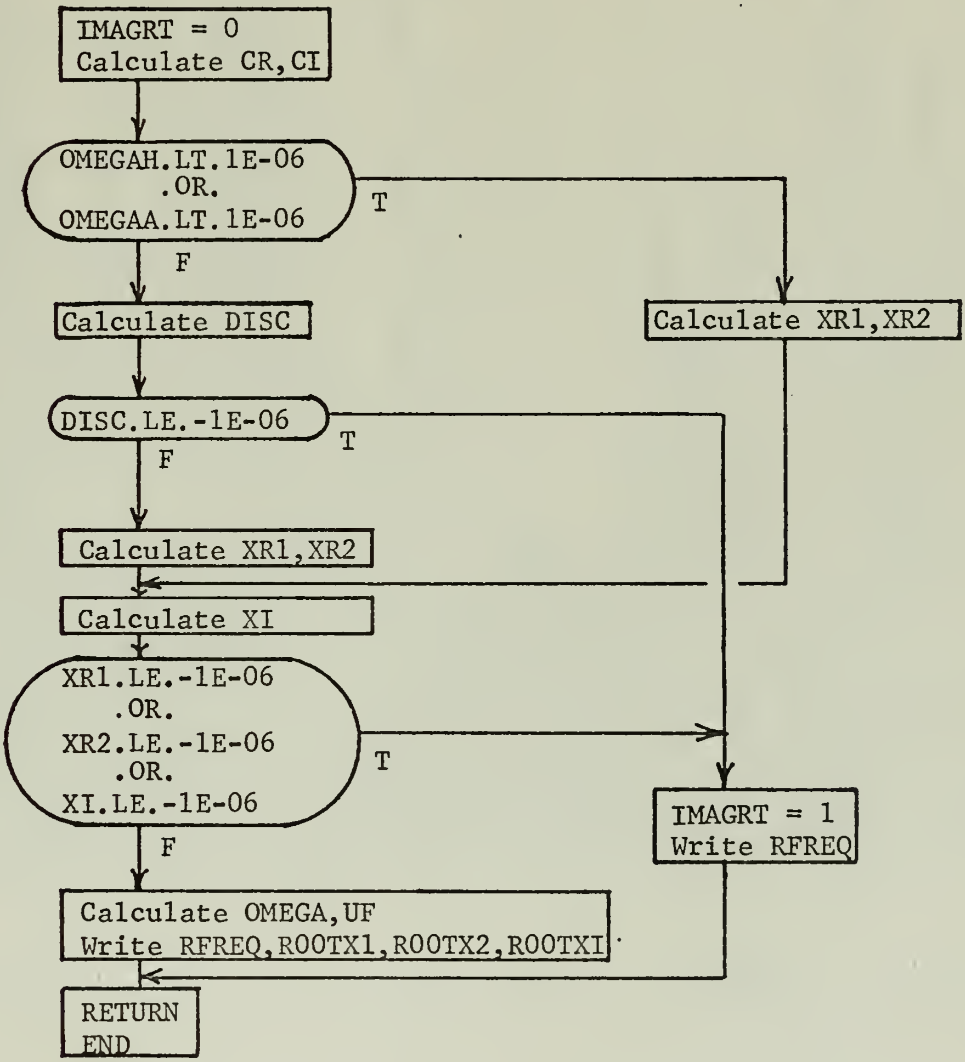

88
94
Define CPR1, CPI1, CPR2,CPI2
IFIN.EQ.1
T
F
Calculate L1,L2, M1, M2
JLINE.NE.ISPLIT .AND. X(K,L)+DSTSTR.LE.1.0
T
F
Calculate L1,L2, M1, M2
IREF = 0
IFIN.EQ.1
T
F
MCOUNT.EQ.0 .AND. X(ILINE,ISWTCH)+DSTSTR.GT.1.0
T
F
140
MCOUNT.EQ.0 .OR. X(ILINE,ISWTCH)+DSTSTR.GT.1.0
T
95
F
Calculate L1,L2, M1, M2
Calculate L1,L2, M1, M2
IREF.EQ.1
T
F
IREF.EQ.1
130
T
F
X(ILINE,ISWTCH)+DSTSTR.GT.1.0
T
F
MCOUNT = MCOUNT+1
JSWICH.NE.1
T
MCOUNT.NE.ISPLIT+1
T
F
110
F
MCOUNT.EQ.IJUNC
T
Calculate L1,L2, M1, M2
F
IREF.EQ.1
T
F
120


MCOUNT = 0
JSWICH.EQ.1
T
F
JSWICH = 2
JSWICH = 1
120
ISPLIT.EQ.-1
T
F
130
JSWICH = 1
ISPLIT.EQ.-1
.OR.
JCOUNT.NE.0
.OR.
DELTA.GT.1E-05
T
F
IREF = 1
NSTPTS.EQ.1
T
F
X(ILINE,ISWTCH)+DSTSTR.GT.1.0
T
95
F
MCOUNT.EQ.0
T
110
F
93
X(ILINE,ISWTCH)+DSTSTR.GT.1.0
T
F
MCOUNT.EQ.0
T
F
MCOUNT-ISPLIT-1
1
-1
0
95
MCOUNT-ISPLIT-2
-1
1
0
X(ILINE,ISWTCH)+DSTSTR.LE.1.0
T
F
140
Calculate DELL1,DELL2,DELM1,DELM2
Calculate L1,L2, M1, M2
DELTA.GT.1E-05
T
F
Calculate L1,L2, M1, M2
IFIN = 1
IREF.EQ.0
T
F
MCOUNT.GT.ISPLIT+1
T
F
RETURN
END

VIII. SUBROUTINE COMPXY



IX. SUBROUTINE FLUTER



```
C     PROGRAM A  --  TWO DEGREE OF FREEDOM FLUTTER                      00000000
C
C     THIS PROGRAM COMPUTES THE FLUTTER FREQUENCY AND FLUTTER VELOCITY  00000001
C     OF A CASCADE OSCILLATING AT LOW AMPLITUDES WITH ARBITRARY STAGGER,00000002
C     PHASE LAG, BLADE SPACING,AND FREQUENCY.  THE CASCADE MUST HAVE    00000003
C     SUPERSONIC LEADING EDGE LOCUS.                                    00000004
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00000010
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00000020
      DIMENSION DATE(3)                                                 00000030
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(20000000040
     1,2),C33I(200,2)                                                   00000050
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH00000060
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00000070
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M100000080
     3,M2       ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL            00000090
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00000100
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00000110
      COMMON/BLK2/L3,L4,M3,M4                                           00000120
      COMMON/BLK3/MUU,RSUBA,XSUBA,HAFREQ,OMEGAA,OMEGAH,INCRE            00000130
      COMMON/BLK4/ROOTX1,ROOTX2,ROOTXI,OMEGA,UF,RFREQ,IMAGRT            00000140
      REAL MUU,INCRE                                                    00000150
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I                         00000160
      LOGICAL FCONST                                                    00000170
      COMMON/FOFX/FCONST                                                00000180
      COMMON/PCOR/KOUNT                                                 00000190
      COMMON/JUNC/IJUNT                                                 00000200
      COMMON/STG/STGR                                                   00000210
      COMMON/FINE/FIN                                                   00000211
      COMMON/IFINE/IFIN,JFIN                                            00000212
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00000220
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00000230
C
C     PRINT NAME OF PROGRAM/DATE OF RUN.                                00000240
C
      READ(5,7)DATE                                                     00000250
    7 FORMAT(3A4)                                                       00000260
      WRITE(6,6)DATE                                                    00000270
    6 FORMAT(1H1,14(/),14(/),37X,51H OSCILLATING CASCADE FLUTTER PROGRAM00000280
     1   ---RUN OF---,3A4,/,1H1)                                        00000290
C
C     PRINT OUT ALL INPUT INFORMATION VIA INPUT.                        00000300
C
      CALL INPUT                                                        00000310
      IZOT=0                                                            00000319
      DO 1005 JTEM=1,50                                                 00000320
      FCONST=.FALSE.                                                    00000330
      DO 1002 ITEM=1,2                                                  00000340
```

```
      ICO=0
      KOUNT=0
      CALL INTIAL
 1000 CONTINUE
C
C     COMPXY COMPUTES THE VALUE FOR X AND Y GIVEN THE PARAMETERS IHAVEP
C    1AND ISWTCH FOR INDICES (RESPECTIVELY).
C
      CALL COMPXY
C
C     THE PROGRAM GOES TO 1 IF POINT IS ON INITIAL MACH LINE.
C
      IF((IHAVEP.EQ.0).AND.(IATNWL.EQ.0))GO TO 1
      IF((IHAVEP.EQ.0).AND.(IATNWL.NE.0))GO TO 2
C
C     TEST HERE IF YOU ARE AT A LOW FOIL PT.IF NOT GO TO NEXT POINT.
C
      IF(IATNWL .EQ. 1 .AND. IHAVEP .EQ. KOUNT) GO TO 3
      IF(IHAVEP.EQ.JLINE)GO TO 3
      GO TO 4
C
C     INITIAL POINT(MACH LINE) CALC. FOR U,V,AND C AT 1.
C
    1 CALL MACHLN
      GO TO 5
C
C     HIGH FOIL   CALCULATION HERE FOR U,V,AND C AT 2.
C
    2 CALL HIFOIL
      GO TO 5
C
C     LOW FOIL PT CALCULATION HERE FOR U,V,AND C AT 3.
C
    3 CALL LOFOIL
      GO TO 5
C
C     GENERAL POINT CALCULATION HERE FOR U,V,AND C AT 4.
C
    4 CALL GENFPT
    5 CONTINUE
      IF(IATNWL .EQ. 1 .AND. IHAVEP .EQ. KOUNT) GO TO 25
      IF(IHAVEP.NE.JLINE)GO TO 100
C
C     TEST IF AT END OF FOIL  IF SO COMPUTE Q AND QUIT.
C
   25 IF(X(IHAVEP+1,ISWTCH)+DSTSTR.GT.FIN) GO TO 101
C
C     TEST IF YOU ARE AT A HI AIRFOIL LINE(RIGHT RUNNING MACH) AND IF
```

```
C    1SO NOTE THIS BY IATNWL=1 AND JLINE=NGDPTS.
C
      IF(ILINE.EQ.NGDPTS)GO TO 102
C
C     INCREMENT FOR NEXT LINES(FIRST OLD ONE(ILINE) BECOMES LAST NEW 1)
C
      ILINE=ILINE+1
      JLINE=JLINE+1
      IF(ILINE.EQ.NGDPTS) IATNWL=1
C
C     SWITCH LINES HERE SO FIRST OLD ONE (ILINE) BECOMES LAST NEW ONE.
C    1ZERO OUT NEW LINE POINT INCREMENT COUNTER.
C
  105 IF(ICO.EQ.1) IJUNC=IJUNC-1
      IF(ISWTCH.EQ.1) GO TO 103
      ISWTCH=1
      JSWTCH=2
      IHAVEP=0
      GO TO 1000
  103 ISWTCH=2
      JSWTCH=1
      IHAVEP=0
      GO TO 1000
C
C     AT 102 SET UP FOR HERE ON IN AT TUNNEL WALL.
C
  102 IATNWL=1
      JLINE=NGDPTS
      KOUNT=NGDPTS-1
      GO TO 105
C
C     AT 100 INCREMENT TO NEXT POINT ALONG PRESENT LINE.
C
  100 IHAVEP=IHAVEP+1
C
C     FINISHED TOP AIRFOIL?  TERMINATE
C
      IF((ICO.EQ.1).AND.(IJUNC.EQ.0)) GO TO 106
C
C     PREVENT UNNECESSARY FLOW FIELD CALCULATION.
C
      IF((ICO.EQ.1).AND.(IHAVEP.GT.IJUNC)) GO TO 105
      GO TO 1000
C
C     IN ZONE 1?  TERMINATE
C
  101 IF((JLINE-1).LT.IJUNC) GO TO 106
      ICO=1
```

```
      GO TO 105                                                         00000990
  106 IF(.NOT.FCONST) GO TO 108                                         00001000
C
C     FINISH PLUNGE MODE CALCULATION.                                   00001010
C
      FACTOR=HDSTRL/(REDFRQ*REDFRQ)                                     00001020
      L1=L1*FACTOR                                                      00001030
      L2=L2*FACTOR                                                      00001040
      M1=M1*FACTOR*2.0                                                  00001050
      M2=M2*FACTOR*2.0                                                  00001060
      GO TO 110                                                         00001070
C
C     FINISH PITCH MODE CALCULATION.                                    00001080
C
  108 FACTOR=DSTSTR/(REDFRQ*REDFRQ)                                     00001090
      L3=L1*FACTOR                                                      00001100
      L4=L2*FACTOR                                                      00001110
      M3=M1*FACTOR*2.0                                                  00001120
      M4=M2*FACTOR*2.0                                                  00001130
  110 FCONST=.TRUE.                                                     00001140
 1002 CONTINUE                                                          00001150
      CALL FLUTER                                                       00001160
      IF(IMAGRT.NE.1) GO TO 200                                         00001161
      IZOT=1                                                            00001162
      GO TO 280                                                         00001163
  200 IF((JTEM.NE.1).OR.(IZOT.NE.1)) GO TO 210                          00001170
      TEST3=ROOTX1-ROOTXI                                               00001180
      TEST4=ROOTX2-ROOTXI                                               00001190
  210 TEST1=ROOTX1-ROOTXI                                               00001200
      TEST2=ROOTX2-ROOTXI                                               00001210
      IF((ABS(TEST1).LE.1E-05).OR.(ABS(TEST2).LE.1E-05)) GO TO 1006     00001220
      IF(TEST1.LT.-1E-05) GO TO 220                                     00001230
      IF(TEST3.LT.-1E-05) GO TO 250                                     00001240
      GO TO 230                                                         00001250
  220 IF(TEST3.GT.1E-05) GO TO 250                                      00001260
  230 IF(TEST2.LT.-1E-05) GO TO 240                                     00001270
      IF(TEST4.LT.-1E-05) GO TO 250                                     00001280
      GO TO 260                                                         00001290
  240 IF(TEST4.GT.1E-05) GO TO 250                                      00001300
      GO TO 260                                                         00001310
  250 RFREQ=RFREQ-INCRE                                                 00001320
      INCRE=INCRE/5.0                                                   00001330
      IF(INCRE.LE.1E-05) GO TO 1006                                     00001340
      GO TO 280                                                         00001350
  260 TEST3=TEST1                                                       00001360
      TEST4=TEST2                                                       00001370
  280 RFREQ=RFREQ+INCRE                                                 00001380
      REDFRQ=2.0/RFREQ                                                  00001381
```

```
 1005 CONTINUE
 1006 WRITE(6,501)OMEGA,UF
  501 FORMAT(////,15X,19H FLUTTER FREQUENCY=,E20.7,15X,18H FLUTTER VELOC
     1ITY=,E20.7)
      WRITE(6,502)JTEM
  502 FORMAT(////,12H ITERATIONS=,I5)
      STOP
      END


      SUBROUTINE INPUT
C
C     SUBROUTINE INPUT READS ALL INPUT.
C     NGRDFN IS THE FINENESS OF GRID NUMBER.
C     FSTRMN IS THE FREESTREAM MACH NUMBER.
C     RTOSPH IS THE RATIO OF SPECIFIC HEATS.
C     REDFRQ IS THE REDUCED DIMENSIONLESS FREQUENCY.
C     TNWDST IS THE DISTANCE BETWEEN AIRFOILS.
C     XSUBO IS THE DISTANCE(DIMENSIONLESS) FROM THE LEADING EDGE TO THE
C    1ELASTIC AXIS.
C     STGANG IS THE CASCADE STAGGER ANGLE IN DEGREES.
C     FAZE IS THE UPPER AIRFOIL PHASE LAG IN DEGREES.
C     DELTA IS THE UPPER AIRFOIL PHASE LAG IN RADIANS.
C     MUU IS THE WING DENSITY PARAMETER.
C     RSUBA IS THE RADIUS OF GYRATION.
C     XSUBA IS THE DISTANCE(DIMENSIONLESS) OF THE CENTER OF GRAVITY
C    1FROM THE ELASTIC AXIS.
C     HAFREQ IS THE RATIO OF THE BENDING NATURAL FREQUENCY OF THE
C     AIRFOIL TO THE TORSIONAL NATURAL FREQUENCY.
C     INCRE IS THE AMOUNT THE REDUCED FREQUENCY IS DECREASED EACH TIME
C    1SQRT(X) IS CALCULATED.
C     OMEGAA IS MUU*RSUBA**2.
C     OMEGAH IS MUU*THE RATIO OF HFREQ TO AFREQ**2.
C     ROOTX IS THE RATIO OF AFREQ TO THE FREQUENCY OF OSCILLATION.
C     HSUBO & ALPHAO ARE THE MAXIMUM AMPLITUDE(DIMENSIONLESS) OF THE
C    1 AIRFOIL PLUNGING & PITCHING OSCILLATION, RESPECTIVELY.  EACH IS
C    2 SET EQUAL TO 1 SINCE THEIR VALUES ARE INDEPENDENT OF THE COMPUTAT
C     FMANGL IS THE MACH ANGLE.
C     XLNGTH IS THE LENGTH OF THE INITIAL MACH LINE.
C     DELTAS IS THE STEP SIZE OF INCREMENTING ALONG THE MACH LINES.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
C
C     NGDPTS IS THE NUMBER OF GRID POINTS(INCL. XSUBO AND TW POINT).
C     DSTSTR IS THE DISTANCE(HORIZONTAL) OF A STREAMLINE FROM ONE GRID
C    1POINT TO THE NEXT STREAMLINE GRID POINT.
C     HDSTRL IS ONE-HALF OF DSTSTR
```

```
C     TRNGLH IS THE HEIGHT OF DELTAS(VERTICAL)
C     U22R IS THE VARIABLE HOLDING THE REAL PART OF U AT THE RIGHTMOST
C    1GRID POINT.THE SECOND INDEX POINTS TO A LINE(MACH) BEING USED.
C     SAME TYPE OF MEANING FOR THE OTHER VARIABLES.
C     AI AND BI ARE USED AS CONSTANTS IN THE DEFINITIONS OF U,V,AND C.
C     COMPUTE ALL VALUESFOR INITIAL MACH LINES INITIAL POINT.
C     SET UP INITIAL INTEGRATION VALUES.
C     ISWTCH IS THE SWITCH VARIABLE FOR CHOOSING THE LINE TO WORK ON.
C     JSWTCH IS THE OPPOSITE OF ISWTCH. IF ISWTCH=1,JSWTCH=2 VICE-VERSA.
C     ILINE CONTAINS THE NUMBER OF POINTS THE LINE EVENTUALLY SHOULD BE.
C     JLINE=ILINE-1 UNLESS AT HI AIRFOILL THEN JLINE=ILINE(JLINE IS TOTA
C    1L NUMBER OF POINTS CONTAINED IN THE OTHER LINE.
C     IHAVE IS A COUNTER TELLING THE NUMBER OF PROCESSED POINTS YOU HAVE
C    1CALCULATED FOR THE LINE YOU ARE WORKING ON.
C     IATNWL=0 SIGNIFIES YOU ARE NOT AT THE HI AIRFOILL YET.IATNWL WILL
C    1BE SET=1 WHEN FINISHED WITH ILINE=NGDPTS(AT HI AIRFOILL).
C
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M1
     3,M2          ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK3/MUU,RSUBA,XSUBA,HAFREQ,OMEGAA,OMEGAH,INCRE
      COMMON/JUNC/IJUNT
      COMMON/STG/STGR
      COMMON/FINE/FIN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      REAL MUU,INCRE
      NAMELIST/NAM1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,STGANG,FAZE
     1,MUU,RSUBA,XSUBA,HAFREQ,INCRE,FIN
      READ(5,NAM1)
      WRITE(6,NAM1)
      FMANGL=ARSIN(1.0/FSTRMN)
      FNGRDN=NGRDFN
      XLNGTH=TNWDST/SIN(FMANGL)
      NGDPTS=NGRDFN
      FNGDPT=NGDPTS
      DELTAS=XLNGTH/FNGDPT
      TRNGLH=TNWDST/FNGDPT
      HDSTRL=DELTAS*COS(FMANGL)
      DSTSTR=HDSTRL*2.0
      STGR=TNWDST*TAN(STGANG*0.1745329E-01)
      NSTPTS=STGR/DSTSTR
      DTSTSR=NSTPTS*DSTSTR
      IF(STGR-DTSTSR.GE.0.50) NSTPTS=NSTPTS+1
      IF(NGDPTS/2.EQ.NSTPTS-1) NSTPTS=NSTPTS-1
```

```
      IF(NGDPTS/2.EQ.NSTPTS+1) NSTPTS=NSTPTS+1                          00002280
      STGR=NSTPTS*DSTSTR                                                00002290
      STGANG=ATAN(STGR/TNWDST)*57.29578                                 00002300
      IJUNT=NGDPTS/2+NSTPTS                                             00002310
      NSTPTS=NSTPTS+1                                                   00002320
      ISPLIT=NGDPTS/2-NSTPTS                                            00002330
      NGDPTS=NGDPTS+1                                                   00002340
      S=SQRT(FSTRMN*FSTRMN-1.0)                                         00002350
      DELTA=FAZE*0.1745329E-01                                          00002360
      OMEGAA=MUU*RSUBA**2                                               00002370
      OMEGAH=MUU*HAFREQ**2                                              00002380
      WRITE(6,5)NGRDFN,FSTRMN,RTOSPH,REDFRQ,TNWDST,XSUBO,STGANG,DSTSTR  00002390
     1,MUU,RSUBA,XSUBA,HAFREQ,FAZE                                      00002400
    5 FORMAT(////,28H GRID FINENESS INPUT NUMBER=,I10,///,24H FREESTREAM00002410
     1 MACH NUMBER=,F20.7,///,25H RATIO OF SPECIFIC HEATS=,F20.7,///,34H00002420
     2 REDUCED(DIMENSIONLESS) FREQUENCY=,F20.7,///,27H DISTANCE BETWEEN 00002430
     3AIRFOILS=,F20.7,///,52H HORIZONTAL POSITION(DIMENSIONLESS) OF ELAS00002440
     4TIC AXIS=,F20.7,///,26H COMPATABLE STAGGER ANGLE=,F20.7,///,9H DEL00002450
     5TA X=,F20.7,///,24H WING DENSITY PARAMETER=,F20.7,///,20H RADIUS O00002460
     6F GYRATION=,F20.7,///,39H DISTANCE(DIMENSIONLESS) FROM EA TO CG=,F00002470
     720.7,///,37H BENDING/TORSIONAL NATURAL FREQUENCY=,F20.7,///,25H UP00002480
     8PER AIRFOIL PHASE LAG=,F20.7)                                     00002490
      WRITE(6,6)                                                        00002500
    6 FORMAT(1H1,10X,80HVALUES OF FREQUENCY RATIO FOR VARIOUS NON-DIMENS00002510
     1IONAL FREQUENCIES OF OSCILLATION)                                 00002520
      RETURN                                                            00002530
      END                                                               00002540

      SUBROUTINE INTIAL                                                 00002550
C
C     INTIAL INITIALIZES ALL FLOW FIELD QUANTITIES.                     00002560
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00002570
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00002580
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(20000002590
     1,2),C33I(200,2)                                                   00002600
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH00002610
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00002620
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M100002630
     3,M2       ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL            00002640
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00002650
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00002660
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I                         00002670
      LOGICAL FCONST                                                    00002680
      COMMON/FOFX/FCONST                                                00002690
      COMMON/JUNC/IJUNT                                                 00002700
      COMMON/IFINE/IFIN,JFIN                                            00002701
```

```
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00002710
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00002720
      IJUNC=IJUNT                                                       00002730
      AI=.5*REDFRQ*DSTSTR                                               00002740
      BI=.25*REDFRQ*(FSTRMN*FSTRMN/(FSTRMN*FSTRMN-1.0))*DSTSTR          00002750
C
C     SET UP INITIAL VALUES FOR U,V,AND C AT (0,0).                     00002760
C
      IF(.NOT. FCONST) GO TO 50                                         00002770
      VRPANL=0.0                                                        00002780
      VIPANL=-REDFRQ/S                                                  00002790
      GO TO 60                                                          00002800
   50 VRPANL=-1.0/S                                                     00002810
      VIPANL=REDFRQ*XSUBO/S                                             00002820
 60   U22R(1,1)=-VRPANL                                                 00002830
      U22I(1,1)=-VIPANL                                                 00002840
      V22R(1,1)=-U22R(1,1)                                              00002850
      V22I(1,1)=-U22I(1,1)                                              00002860
      C22R(1,1)=-U22R(1,1)                                              00002870
      C22I(1,1)=-U22I(1,1)                                              00002880
      X(1,1)=0.0                                                        00002890
      Y(1,1)=0.0                                                        00002900
      IF((ISPLIT.NE.-1).AND.(DELTA.LE.1E-05)) GO TO 63                  00002910
      L1=-C22R(1,1)                                                     00002920
      L2=-C22I(1,1)                                                     00002930
      M1=XSUBO*C22R(1,1)                                                00002940
      M2=XSUBO*C22I(1,1)                                                00002950
      GO TO 64                                                          00002960
   63 L1=-2.0*C22R(1,1)                                                 00002970
      L2=-2.0*C22I(1,1)                                                 00002980
      M1=2.0*XSUBO*C22R(1,1)                                            00002990
      M2=2.0*XSUBO*C22I(1,1)                                            00003000
   64 IF(DELTA.LE.1E-05) GO TO 65                                       00003010
      IF(.NOT.FCONST) GO TO 150                                         00003020
      VRPANL=REDFRQ*SIN(DELTA)/S                                        00003030
      VIPANL=-REDFRQ*COS(DELTA)/S                                       00003040
      GO TO 160                                                         00003050
  150 VRPANL=-(COS(DELTA)+REDFRQ*SIN(DELTA)*XSUBO)/S                    00003060
      VIPANL=-(SIN(DELTA)-REDFRQ*COS(DELTA)*XSUBO)/S                    00003070
  160 U33R(1,1)=VRPANL                                                  00003080
      U33I(1,1)=VIPANL                                                  00003090
      V33R(1,1)=U33R(1,1)                                               00003100
      V33I(1,1)=U33I(1,1)                                               00003110
      C33R(1,1)=-U33R(1,1)                                              00003120
      C33I(1,1)=-U33I(1,1)                                              00003130
      IF(ISPLIT.EQ.-1) GO TO 65                                         00003140
      L1=L1+C33R(1,1)*COS(DELTA)+C33I(1,1)*SIN(DELTA)                   00003150
      L2=L2+C33I(1,1)*COS(DELTA)-C33R(1,1)*SIN(DELTA)                   00003160
```

```
      M1=M1-XSUBO*(C33R(1,1)*COS(DELTA)+C33I(1,1)*SIN(DELTA))
      M2=M2-XSUBO*(C33I(1,1)*COS(DELTA)-C33R(1,1)*SIN(DELTA))
   65 ISWTCH=2
      JSWTCH=1
      ISWICH=1
      JSWICH=1
      IFIN=0
      JFIN=0
      ILINE=2
      JLINE=1
      IHAVEP=0
      IATNWL=0
      JCOUNT=0
      LCOUNT=0
      MCOUNT=1
      RETURN
      END


      SUBROUTINE MACHLN
C
C     MACHLN COMPUTES THE VALUES OF U,V,AND C ALONG THE INITIAL MACH
C    1LINE AT THE GIVEN X VALUE OF THE MACH LINE.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M1
     3,M2         ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I
      LOGICAL FCONST
      COMMON/FOFX/FCONST
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      ICOUNT=JLINE
      IF(ICOUNT.LT.IJUNC) GO TO 10
      JCOUNT=JCOUNT+1
      KCOUNT=JCOUNT-1
      IF(KCOUNT.NE.0) GO TO 5
      UR=0.0
      UI=0.0
      VR=0.0
      VI=0.0
```

```
      CR=0.0
      CI=0.0
      GO TO 7
    5 UR=U22R(KCOUNT,3)
      UI=U22I(KCOUNT,3)
      VR=V22R(KCOUNT,3)
      VI=V22I(KCOUNT,3)
      CR=C22R(KCOUNT,3)
      CI=C22I(KCOUNT,3)
    7 K12R=UR+CR+AI*UI
      K12I=-AI*UR+UI+CI
      K34R=U22R(1,JSWTCH)-V22R(1,JSWTCH)+BI*(U22I(1,JSWTCH)-C22I(1,JSWTC
     1H))
      K34I=U22I(1,JSWTCH)-V22I(1,JSWTCH)-BI*(U22R(1,JSWTCH)-C22R(1,JSWTC
     1H))
      K56R=U22R(JCOUNT,3)+V22R(JCOUNT,3)+BI*(U22I(JCOUNT,3)-C22I(JCOUNT,
     13))
      K56I=U22I(JCOUNT,3)+V22I(JCOUNT,3)-BI*(U22R(JCOUNT,3)-C22R(JCOUNT,
     13))
      G1=.5*(K34R+K56R)
      G2=BI*K12I
      G3=1.0-AI*BI
      G4=.5*(K34I+K56I)
      G5=BI*K12R
      G6=2.0*BI
      G7=G3*G3+G6*G6
      G8=G1-G2
      G9=G4+G5
      U22R(1,ISWTCH)=(G8*G3+G9*G6)/G7
      U22I(1,ISWTCH)=(-G8*G6+G9*G3)/G7
      V22R(1,ISWTCH)=.5*(K56R-K34R)
      V22I(1,ISWTCH)=.5*(K56I-K34I)
      C22R(1,ISWTCH)=K12R-U22R(1,ISWTCH)                  +U22I(1,ISWTCH)*A
     1I
      C22I(1,ISWTCH)=K12I-U22I(1,ISWTCH)                  -U22R(1,ISWTCH)*A
     1I
      GO TO 503
   10 W=FSTRMN*FSTRMN
      T=FSTRMN*FSTRMN-1.0
      IF(.NOT. FCONST) GO TO 20
      VRPANL=0.0
      VIPANL=-REDFRQ/S
      GO TO 30
   20 VRPANL=-1.0/S
      VIPANL=REDFRQ*XSUBO/S
 30   U=COS(REDFRQ*(W/T)*X(1,ISWTCH))
      V=   SIN(REDFRQ*(W/T)*X(1,ISWTCH))
C
```

```
C     HIERARCHY W,T,S,U,V                                               00004060
C
      U22R(1,ISWTCH)=-VRPANL*U-VIPANL*V                                 00004070
      U22I(1,ISWTCH)=-VIPANL*U+VRPANL*V                                 00004080
      V22R(1,ISWTCH)=-U22R(1,ISWTCH)                                    00004090
      V22I(1,ISWTCH)=-U22I(1,ISWTCH)                                    00004100
      C22R(1,ISWTCH)=-U22R(1,ISWTCH)                                    00004110
      C22I(1,ISWTCH)=-U22I(1,ISWTCH)                                    00004120
      IF((ICOUNT.NE.ISPLIT).OR.(DELTA.GT.1E-05)) GO TO 500              00004130
      U22R(1,3)=-U22R(1,ISWTCH)                                         00004140
      U22I(1,3)=-U22I(1,ISWTCH)                                         00004150
      V22R(1,3)= V22R(1,ISWTCH)                                         00004160
      V22I(1,3)= V22I(1,ISWTCH)                                         00004170
      C22R(1,3)=-C22R(1,ISWTCH)                                         00004180
      C22I(1,3)=-C22I(1,ISWTCH)                                         00004190
      GO TO 503                                                         00004200
  500 IF((DELTA.LE.1E-05).OR.(JLINE.GT.ISPLIT)) GO TO 503               00004210
      IF(.NOT.FCONST) GO TO 120                                         00004220
      VRPANL=REDFRQ*SIN(DELTA)/S                                        00004230
      VIPANL=-REDFRQ*COS(DELTA)/S                                       00004240
      GO TO 130                                                         00004250
  120 VRPANL=-(COS(DELTA)+REDFRQ*SIN(DELTA)*XSUBO)/S                    00004260
      VIPANL=-(SIN(DELTA)-REDFRQ*COS(DELTA)*XSUBO)/S                    00004270
  130 U33R(1,ISWTCH)=VRPANL*U+VIPANL*V                                  00004280
      U33I(1,ISWTCH)=VIPANL*U-VRPANL*V                                  00004290
      V33R(1,ISWTCH)=U33R(1,ISWTCH)                                     00004300
      V33I(1,ISWTCH)=U33I(1,ISWTCH)                                     00004310
      C33R(1,ISWTCH)=-U33R(1,ISWTCH)                                    00004320
      C33I(1,ISWTCH)=-U33I(1,ISWTCH)                                    00004330
      IF(ICOUNT.NE.ISPLIT) GO TO 503                                    00004340
      U22R(1,3)=U33R(1,ISWTCH)                                          00004350
      U22I(1,3)=U33I(1,ISWTCH)                                          00004360
      V22R(1,3)=V33R(1,ISWTCH)                                          00004370
      V22I(1,3)=V33I(1,ISWTCH)                                          00004380
      C22R(1,3)=C33R(1,ISWTCH)                                          00004390
      C22I(1,3)=C33I(1,ISWTCH)                                          00004400
  503 CONTINUE                                                          00004410
      RETURN                                                            00004420
      END                                                               00004430

      SUBROUTINE HIFOIL                                                 00004440
C
C     HIFOIL COMPUTES U,V,AND C AT AN UPPER AIRFOIL POINT.              00004450
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00004460
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00004470
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH00004480
```

```
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00004490
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M100004500
     3,M2        ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL           00004510
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00004520
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00004530
      COMMON/PCOR/KOUNT                                                 00004540
      COMMON/STG/STGR                                                   00004550
      COMMON/JUNC/IJUNT                                                 00004560
      COMMON/IFINE/IFIN,JFIN                                            00004561
      LOGICAL FCONST                                                    00004570
      COMMON/FOFX/FCONST                                                00004580
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00004590
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00004600
      TOPCRD=X(1,ISWTCH)-STGR                                           00004610
      IF(JLINE.EQ.NGDPTS-1) IATNWL=0                                    00004620
      IF(ICOUNT.EQ.0) GO TO 80                                          00004630
      IF(JCOUNT.NE.0) GO TO 75                                          00004640
      K12R=0.0                                                          00004650
      K12I=0.0                                                          00004660
      J=1                                                               00004670
      GO TO 80                                                          00004680
   75 K12R=U22R(JCOUNT,3)+C22R(JCOUNT,3)+AI*U22I(JCOUNT,3)              00004690
      K12I=-AI*U22R(JCOUNT,3)+U22I(JCOUNT,3)+C22I(JCOUNT,3)             00004700
      J=1                                                               00004710
   80 IF(ICOUNT.NE.0) GO TO 85                                          00004720
      I=1                                                               00004730
      J=2                                                               00004740
      K12R=U22R(I    ,JSWTCH)+C22R(I    ,JSWTCH)+AI*U22I(I    ,JSWTCH)  00004750
      K12I=-AI*U22R(I    ,JSWTCH)+U22I(I    ,JSWTCH)+C22I(I    ,JSWTCH) 00004760
   85 IF(.NOT. FCONST) GO TO 999                                        00004770
      V22R(1,ISWTCH)=REDFRQ*SIN(DELTA)/S                                00004780
      V22I(1,ISWTCH)=-REDFRQ*COS(DELTA)/S                               00004790
      GO TO 90                                                          00004800
  999 V22R(1,ISWTCH)=-(COS(DELTA)-REDFRQ*SIN(DELTA)*(TOPCRD-XSUBO))/S   00004810
      V22I(1,ISWTCH)=-(SIN(DELTA)+REDFRQ*COS(DELTA)*(TOPCRD-XSUBO))/S   00004820
 90   K56R=V22R(1    ,ISWTCH)                                           00004830
      K56I=V22I(1    ,ISWTCH)                                           00004840
 91   K34R=U22R(J    ,JSWTCH)-V22R(J    ,JSWTCH)+BI*(U22I(J    ,JSWTCH)-00004850
     1C22I(J    ,JSWTCH))                                               00004860
      K34I=U22I(J    ,JSWTCH)-V22I(J    ,JSWTCH)-BI*(U22R(J    ,JSWTCH)-00004870
     1C22R(J    ,JSWTCH))                                               00004880
 92   G1=1.0-AI*BI                                                      00004890
      G2=2.0*BI                                                         00004900
      G3=G1*G1+G2*G2                                                    00004910
      G4=K56R+K34R-BI*K12I                                              00004920
      G5=K56I+K34I+BI*K12R                                              00004930
      U22R(1    ,ISWTCH)=(G4*G1+G5*G2)/G3                               00004940
      U22I(1    ,ISWTCH)=(-G4*G2+G5*G1)/G3                              00004950
```

```
      C22R(1     ,ISWTCH)=K12R-U22R(1     ,ISWTCH)+U22I(1     ,ISWTCH)*AI  00004960
      C22I(1     ,ISWTCH)=K12I-U22I(1     ,ISWTCH)-U22R(1     ,ISWTCH)*AI  00004970
      CPR1=C22R(1,JSWTCH)*COS(DELTA)+C22I(1,JSWTCH)*SIN(DELTA)              00004980
      CPI1=C22I(1,JSWTCH)*COS(DELTA)-C22R(1,JSWTCH)*SIN(DELTA)              00004990
      CPR2=C22R(1,ISWTCH)*COS(DELTA)+C22I(1,ISWTCH)*SIN(DELTA)              00005000
      CPI2=C22I(1,ISWTCH)*COS(DELTA)-C22R(1,ISWTCH)*SIN(DELTA)              00005010
      IF(JFIN.EQ.1) GO TO 130                                               00005011
C
C     AT START OF ZONE AND CLOSE TO END?  SKIP LIFT CALCULATION.            00005020
C
      IF((LCOUNT.EQ.0).AND.(TOPCRD+DSTSTR.GT.1.0)) GO TO 130                00005030
C
C     AT START OF ZONE OR END OF AIRFOIL?  START/STOP INTEGRATION.          00005040
C
      IF((LCOUNT.EQ.0).OR.(TOPCRD+DSTSTR.GT.1.0)) GO TO 95                  00005050
      L1=L1+2.0*CPR2                                                        00005060
      L2=L2+2.0*CPI2                                                        00005070
      M1=M1+2.0*CPR2*(TOPCRD-XSUBO)                                         00005080
      M2=M2+2.0*CPI2*(TOPCRD-XSUBO)                                         00005090
      GO TO 100                                                             00005100
   95 L1=L1+CPR2                                                            00005110
      L2=L2+CPI2                                                            00005120
      M1=M1+CPR2*(TOPCRD-XSUBO)                                             00005130
      M2=M2+CPI2*(TOPCRD-XSUBO)                                             00005140
  100 LCOUNT=LCOUNT+1                                                       00005150
C
C     SET UP FOR END OF ZONE CHECK.                                         00005160
C
      IF(ISWICH.NE.1) GO TO 105                                             00005170
      IF(LCOUNT.EQ.IJUNT) GO TO 110                                         00005180
      GO TO 120                                                             00005190
  105 IF(LCOUNT.NE.JCOUNT) GO TO 120                                        00005200
C
C     AT END OF ZONE, LINEARLY EXTRAPOLATE LAST VALUE AND STOP INTEGRAL.00005210
C
  110 L1=L1+2.0*CPR2-CPR1                                                   00005220
      L2=L2+2.0*CPI2-CPI1                                                   00005230
      M1=M1+(2.0*CPR2-CPR1)*(TOPCRD+DSTSTR-XSUBO)                           00005240
      M2=M2+(2.0*CPI2-CPI1)*(TOPCRD+DSTSTR-XSUBO)                           00005250
C
C     ZERO COUNTER AND SWITCH ZONE MARKERS.                                 00005260
C
      LCOUNT=0                                                              00005270
      IF(ISWICH.EQ.1) GO TO 115                                             00005280
      ISWICH=1                                                              00005290
      GO TO 120                                                             00005300
  115 ISWICH=2                                                              00005310
C
```

```
C     ALPHA=90-BETA?  ELIMINATE ODD ZONE.                               00005320
C
      IF(JCOUNT.EQ.0) ISWICH=1                                          00005330
  120 CONTINUE                                                          00005340
      IF(TOPCRD+DSTSTR.LE.1.0) GO TO 130                                00005350
C
C     COMPUTE VALUES OF C22 AT TOPCRD=1.                                00005360
C
      DELCR=C22R(1,ISWTCH)+(C22R(1,ISWTCH)-C22R(1,JSWTCH))*(1.0-X(1,ISWT00005370
     1CH)+STGR)/DSTSTR                                                  00005380
      DELCI=C22I(1,ISWTCH)+(C22I(1,ISWTCH)-C22I(1,JSWTCH))*(1.0-X(1,ISWT00005390
     1CH)+STGR)/DSTSTR                                                  00005400
      DELCPR=DELCR*COS(DELTA)+DELCI*SIN(DELTA)                          00005410
      DELCPI=DELCI*COS(DELTA)-DELCR*SIN(DELTA)                          00005420
C
C     COMPUTE CHANGE IN LIFT AND MOMENT DUE TO UNDERSHOOT.              00005430
C
      DELL1=(CPR2+DELCPR)*(1.0-TOPCRD)/DSTSTR                           00005440
      DELL2=(CPI2+DELCPI)*(1.0-TOPCRD)/DSTSTR                           00005450
      DELM1=(CPR2*(TOPCRD-XSUBO)+DELCPR*(1.0-XSUBO))*(1.0-TOPCRD)/DSTSTR00005460
      DELM2=(CPI2*(TOPCRD-XSUBO)+DELCPI*(1.0-XSUBO))*(1.0-TOPCRD)/DSTSTR00005470
      L1=L1+DELL1                                                       00005480
      L2=L2+DELL2                                                       00005490
      M1=M1+DELM1                                                       00005500
      M2=M2+DELM2                                                       00005510
      JFIN=1                                                            00005511
  130 CONTINUE                                                          00005520
      ICOUNT=0                                                          00005530
      RETURN                                                            00005540
      END                                                               00005550


      SUBROUTINE GENFPT                                                 00005560
C
C     GENFPT COMPUTES U,V,AND C AT A GENERAL FIELD POINT.               00005570
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00005580
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00005590
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(20000005600
     1,2),C33I(200,2)                                                   00005610
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH00005620
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00005630
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M100005640
     3,M2        ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL           00005650
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00005660
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00005670
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I                         00005680
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00005690
```

```
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      I=IHAVEP+1
      IF(IATNWL .EQ. 0) GO TO 10
      K12R=U22R(I,JSWTCH)+C22R(I,JSWTCH)+AI*U22I(I,JSWTCH)
      K12I=-AI*U22R(I,JSWTCH)+U22I(I,JSWTCH)+C22I(I,JSWTCH)
      K34R=U22R(I+1,JSWTCH)-V22R(I+1,JSWTCH)+BI*(U22I(I+1,JSWTCH)-C22I(I
     1+1,JSWTCH))
      K34I=U22I(I+1,JSWTCH)-V22I(I+1,JSWTCH)-BI*(U22R(I+1,JSWTCH)-C22R(I
     1+1,JSWTCH))
      GO TO 12
 10   K12R=U22R(IHAVEP,JSWTCH)+C22R(IHAVEP,JSWTCH)+AI*U22I(IHAVEP,JSWTCH
     1)
      K12I=-AI*U22R(IHAVEP,JSWTCH)+U22I(IHAVEP,JSWTCH)+C22I(IHAVEP,JSWTC
     1H)
      K34R=U22R(IHAVEP+1,JSWTCH)-V22R(IHAVEP+1,JSWTCH)+BI*(U22I(IHAVEP+1
     1,JSWTCH)-C22I(IHAVEP+1,JSWTCH))
      K34I=U22I(IHAVEP+1,JSWTCH)-V22I(IHAVEP+1,JSWTCH)-BI*(U22R(IHAVEP+1
     1,JSWTCH)-C22R(IHAVEP+1,JSWTCH))
 12   K56R=U22R(IHAVEP,ISWTCH)+V22R(IHAVEP,ISWTCH)+BI*(U22I(IHAVEP,ISWTC
     1H)-C22I(IHAVEP,ISWTCH))
      K56I=U22I(IHAVEP,ISWTCH)+V22I(IHAVEP,ISWTCH)-BI*(U22R(IHAVEP,ISWTC
     1H)-C22R(IHAVEP,ISWTCH))
      G1=.5*(K34R+K56R)
      G2=BI*K12I
      G3=1.0-AI*BI
      G4=.5*(K34I+K56I)
      G5=BI*K12R
      G6=2.0*BI
      G7=G3*G3+G6*G6
      G8=G1-G2
      G9=G4+G5
      U22R(I,ISWTCH)=(G8*G3+G9*G6)/G7
      U22I(I,ISWTCH)=(-G8*G6+G9*G3)/G7
      V22R(I,ISWTCH)=.5*(K56R-K34R)
      V22I(I,ISWTCH)=.5*(K56I-K34I)
      C22R(I,ISWTCH)=K12R-U22R(I,ISWTCH)                     +U22I(I,ISWTCH)*A
     1I
      C22I(I,ISWTCH)=K12I-U22I(I,ISWTCH)                     -U22R(I,ISWTCH)*A
     1I
      IF((ICOUNT.NE.ISPLIT).OR.(DELTA.GT.1E-05)) GO TO 450
      U22R(I,3)=-U22R(I,ISWTCH)
      U22I(I,3)=-U22I(I,ISWTCH)
      V22R(I,3)= V22R(I,ISWTCH)
      V22I(I,3)= V22I(I,ISWTCH)
      C22R(I,3)=-C22R(I,ISWTCH)
      C22I(I,3)=-C22I(I,ISWTCH)
      GO TO 500
  450 IF((DELTA.LE.1E-05).OR.(JLINE.GT.ISPLIT)) GO TO 500
```

```
      K12R=U33R(IHAVEP,JSWTCH)+C33R(IHAVEP,JSWTCH)+AI*U33I(IHAVEP,JSWTCH
     1)
      K12I=-AI*U33R(IHAVEP,JSWTCH)+U33I(IHAVEP,JSWTCH)+C33I(IHAVEP,JSWTC
     1H)
      K34R=U33R(IHAVEP,ISWTCH)-V33R(IHAVEP,ISWTCH)+BI*(U33I(IHAVEP,ISWTC
     1H)-C33I(IHAVEP,ISWTCH))
      K34I=U33I(IHAVEP,ISWTCH)-V33I(IHAVEP,ISWTCH)-BI*(U33R(IHAVEP,ISWTC
     1H)-C33R(IHAVEP,ISWTCH))
      K56R=U33R(IHAVEP+1,JSWTCH)+V33R(IHAVEP+1,JSWTCH)+BI*(U33I(IHAVEP+1
     1,JSWTCH)-C33I(IHAVEP+1,JSWTCH))
      K56I=U33I(IHAVEP+1,JSWTCH)+V33I(IHAVEP+1,JSWTCH)-BI*(U33R(IHAVEP+1
     1,JSWTCH)-C33R(IHAVEP+1,JSWTCH))
      G1=.5*(K34R+K56R)
      G2=BI*K12I
      G4=.5*(K34I+K56I)
      G5=BI*K12R
      G8=G1-G2
      G9=G4+G5
      U33R(I,ISWTCH)=(G8*G3+G9*G6)/G7
      U33I(I,ISWTCH)=(-G8*G6+G9*G3)/G7
      V33R(I,ISWTCH)=0.5*(K56R-K34R)
      V33I(I,ISWTCH)=0.5*(K56I-K34I)
      C33R(I,ISWTCH)=K12R-U33R(I,ISWTCH)+U33I(I,ISWTCH)*AI
      C33I(I,ISWTCH)=K12I-U33I(I,ISWTCH)-U33R(I,ISWTCH)*AI
      IF(JLINE.NE.ISPLIT) GO TO 500
      U22R(I,3)=U33R(I,ISWTCH)
      U22I(I,3)=U33I(I,ISWTCH)
      V22R(I,3)=V33R(I,ISWTCH)
      V22I(I,3)=V33I(I,ISWTCH)
      C22R(I,3)=C33R(I,ISWTCH)
      C22I(I,3)=C33I(I,ISWTCH)
  500 CONTINUE
      RETURN
      END


      SUBROUTINE LOFOIL
C
C     LOFOIL COMPUTES THE VALUES OF U,V,AND C AT A LOWER AIRFOIL POINT.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M1
     3,M2         ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
```

```
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00006620
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00006630
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I                         00006640
      COMMON/PCOR/KOUNT                                                 00006650
      COMMON/IFINE/IFIN,JFIN                                            00006651
      LOGICAL FCONST                                                    00006660
      COMMON/FOFX/FCONST                                                00006670
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00006680
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00006690
      IF(.NOT. FCONST) GO TO 999                                        00006700
      V22R(ILINE,ISWTCH)=0.0                                            00006710
      V22I(ILINE,ISWTCH)=-REDFRQ/SQRT(FSTRMN*FSTRMN-1.0)                00006720
      GO TO 90                                                          00006730
  999 V22R(ILINE,ISWTCH)=-1.0/SQRT(FSTRMN*FSTRMN-1.0)                   00006740
      V22I(ILINE,ISWTCH)=(-REDFRQ*(X(IHAVEP+1,ISWTCH)-XSUBO))/SQRT(FSTRM00006750
     1N*FSTRMN-1.0)                                                     00006760
 90   K34R=V22R(ILINE,ISWTCH)                                           00006770
      K34I=V22I(ILINE,ISWTCH)                                           00006780
      K12R=U22R(JLINE,JSWTCH)+C22R(JLINE,JSWTCH)+AI*U22I(JLINE,JSWTCH)  00006790
      K12I=-AI*U22R(JLINE,JSWTCH)+U22I(JLINE,JSWTCH)+C22I(JLINE,JSWTCH) 00006800
      IF (IATNWL .EQ. 0) GO TO 91                                       00006810
      K56R=U22R(KOUNT,ISWTCH) +V22R(KOUNT,ISWTCH)+BI*(U22I(KOUNT,ISWTCH)00006820
     1-C22I(KOUNT,ISWTCH))                                              00006830
      K56I=U22I(KOUNT,ISWTCH)+V22I(KOUNT,ISWTCH)-BI*(U22R(KOUNT,ISWTCH)-00006840
     1C22R(KOUNT,ISWTCH))                                               00006850
      GO TO 92                                                          00006860
 91   K56R=U22R(JLINE,ISWTCH)+V22R(JLINE,ISWTCH)+BI*(U22I(JLINE,ISWTCH)-00006870
     1C22I(JLINE,ISWTCH))                                               00006880
      K56I=U22I(JLINE,ISWTCH)+V22I(JLINE,ISWTCH)-BI*(U22R(JLINE,ISWTCH)-00006890
     1C22R(JLINE,ISWTCH))                                               00006900
 92   G1=1.0-AI*BI                                                      00006910
      G2=2.0*BI                                                         00006920
      G3=G1*G1+G2*G2                                                    00006930
      G4=K56R-K34R-BI*K12I                                              00006940
      G5=K56I-K34I+BI*K12R                                              00006950
      U22R(ILINE,ISWTCH)=(G4*G1+G5*G2)/G3                               00006960
      U22I(ILINE,ISWTCH)=(-G4*G2+G5*G1)/G3                              00006970
      C22R(ILINE,ISWTCH)=K12R-U22R(ILINE,ISWTCH)+U22I(ILINE,ISWTCH)*AI  00006980
      C22I(ILINE,ISWTCH)=K12I-U22I(ILINE,ISWTCH)-U22R(ILINE,ISWTCH)*AI  00006990
      K=ILINE                                                           00007000
      L=ISWTCH                                                          00007010
      IF((ICOUNT.NE.ISPLIT).OR.(DELTA.GT.1E-05)) GO TO 80               00007020
      U22R(K,3)=-U22R(K,L)                                              00007030
      U22I(K,3)=-U22I(K,L)                                              00007040
      V22R(K,3)= V22R(K,L)                                              00007050
      V22I(K,3)= V22I(K,L)                                              00007060
      C22R(K,3)=-C22R(K,L)                                              00007070
      C22I(K,3)=-C22I(K,L)                                              00007080
```

```
      GO TO 94                                                          00007090
   80 IF((DELTA.LE.1E-05).OR.(JLINE.GT.ISPLIT)) GO TO 94                00007100
      IF(.NOT.FCONST) GO TO 84                                          00007110
      V33R(K,L)=REDFRQ*SIN(DELTA)/S                                     00007120
      V33I(K,L)=-REDFRQ*COS(DELTA)/S                                    00007130
      GO TO 86                                                          00007140
   84 V33R(K,L)=-(COS(DELTA)-REDFRQ*SIN(DELTA)*(X(K,L)-XSUBO))/S        00007150
      V33I(K,L)=-(SIN(DELTA)+REDFRQ*COS(DELTA)*(X(K,L)-XSUBO))/S        00007160
   86 K12R=U33R(JLINE,JSWTCH)+C33R(JLINE,JSWTCH)+AI*U33I(JLINE,JSWTCH)  00007170
      K12I=U33I(JLINE,JSWTCH)+C33I(JLINE,JSWTCH)-AI*U33R(JLINE,JSWTCH)  00007180
      K34R=U33R(JLINE,L)-V33R(JLINE,L)+BI*(U33I(JLINE,L)-C33I(JLINE,L)) 00007190
      K34I=U33I(JLINE,L)-V33I(JLINE,L)-BI*(U33R(JLINE,L)-C33R(JLINE,L)) 00007200
      K56R=V33R(K,L)                                                    00007210
      K56I=V33I(K,L)                                                    00007220
      G4=K56R+K34R-BI*K12I                                              00007230
      G5=K56I+K34I+BI*K12R                                              00007240
      U33R(K,L)=(G4*G1+G5*G2)/G3                                        00007250
      U33I(K,L)=-(G4*G2-G5*G1)/G3                                       00007260
      C33R(K,L)=K12R-U33R(K,L)+U33I(K,L)*AI                             00007270
      C33I(K,L)=K12I-U33I(K,L)-U33R(K,L)*AI                             00007280
      IF(JLINE.NE.ISPLIT) GO TO 88                                      00007290
      U22R(K,3)=U33R(K,L)                                               00007300
      U22I(K,3)=U33I(K,L)                                               00007310
      V22R(K,3)=V33R(K,L)                                               00007320
      V22I(K,3)=V33I(K,L)                                               00007330
      C22R(K,3)=C33R(K,L)                                               00007340
      C22I(K,3)=C33I(K,L)                                               00007350
   88 CPR2=C33R(K,L)*COS(DELTA)+C33I(K,L)*SIN(DELTA)                    00007360
      CPI2=C33I(K,L)*COS(DELTA)-C33R(K,L)*SIN(DELTA)                    00007370
      CPR1=C33R(JLINE,JSWTCH)*COS(DELTA)+C33I(JLINE,JSWTCH)*SIN(DELTA)  00007380
      CPI1=C33I(JLINE,JSWTCH)*COS(DELTA)-C33R(JLINE,JSWTCH)*SIN(DELTA)  00007390
      IF(IFIN.EQ.1) GO TO 140                                           00007391
      L1=L1+2.0*CPR2                                                    00007400
      L2=L2+2.0*CPI2                                                    00007410
      M1=M1+2.0*CPR2*(X(K,L)-XSUBO)                                     00007420
      M2=M2+2.0*CPI2*(X(K,L)-XSUBO)                                     00007430
      IF((JLINE.NE.ISPLIT).AND.(X(K,L)+DSTSTR.LE.1.0)) GO TO 94         00007440
      L1=L1+2.0*CPR2-CPR1                                               00007450
      L2=L2+2.0*CPI2-CPI1                                               00007460
      M1=M1+(2.0*CPR2-CPR1)*(X(K,L)+DSTSTR-XSUBO)                       00007470
      M2=M2+(2.0*CPI2-CPI1)*(X(K,L)+DSTSTR-XSUBO)                       00007480
   94 IREF=0                                                            00007490
      IF(IFIN.EQ.1) GO TO 140                                           00007491
C
C     AT START OF ZONE AND CLOSE TO END?  SKIP LIFT CALCULATION.        00007500
C
      IF((MCOUNT.EQ.0).AND.(X(ILINE,ISWTCH)+DSTSTR.GT.1.0)) GO TO 140   00007510
C
```

```
C     AT START OF ZONE OR END OF AIRFOIL?  START/STOP INTEGRATION.      00007520
C
      IF((MCOUNT.EQ.0).OR.(X(ILINE,ISWTCH)+DSTSTR.GT.1.0)) GO TO 95     00007530
   93 L1=L1-2.0*C22R(ILINE,ISWTCH)                                      00007540
      L2=L2-2.0*C22I(ILINE,ISWTCH)                                      00007550
      M1=M1-2.0*C22R(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)              00007560
      M2=M2-2.0*C22I(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)              00007570
      IF(IREF.EQ.1) GO TO 130                                           00007580
      GO TO 100                                                         00007590
   95 L1=L1-C22R(ILINE,ISWTCH)                                          00007600
      L2=L2-C22I(ILINE,ISWTCH)                                          00007610
      M1=M1-C22R(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)                  00007620
      M2=M2-C22I(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)                  00007630
      IF(IREF.EQ.1) GO TO 130                                           00007640
      IF(X(ILINE,ISWTCH)+DSTSTR.GT.1.0) GO TO 120                       00007650
  100 MCOUNT=MCOUNT+1                                                   00007660
C
C     SET UP FOR END OF ZONE CHECK.                                     00007670
C
      IF(JSWICH.NE.1) GO TO 105                                         00007680
      IF(MCOUNT.EQ.IJUNC) GO TO 110                                     00007690
      GO TO 120                                                         00007700
  105 IF(MCOUNT.NE.(ISPLIT+1)) GO TO 120                                00007710
C
C     AT END OF ZONE, LINEARLY EXTRAPOLATE LAST VALUE AND STOP INTEGRAL.00007720
C
  110 L1=L1-2.0*C22R(ILINE,ISWTCH)+C22R(JLINE,JSWTCH)                   00007730
      L2=L2-2.0*C22I(ILINE,ISWTCH)+C22I(JLINE,JSWTCH)                   00007740
      M1=M1-(2.0*C22R(ILINE,ISWTCH)-C22R(JLINE,JSWTCH))*(X(ILINE,ISWTCH)00007750
     1+DSTSTR-XSUBO)                                                    00007760
      M2=M2-(2.0*C22I(ILINE,ISWTCH)-C22I(JLINE,JSWTCH))*(X(ILINE,ISWTCH)00007770
     1+DSTSTR-XSUBO)                                                    00007780
C
C     IN ZONE 1?  GET LIFT & MOMENT OF TOP AIRFOIL.                     00007790
C
      IF(IREF.EQ.1) GO TO 93                                            00007800
C
C     ZERO COUNTER AND SWITCH ZONE MARKERS.                             00007810
C
      MCOUNT=0                                                          00007820
      IF(JSWICH.EQ.1) GO TO 115                                         00007830
      JSWICH=1                                                          00007840
      GO TO 120                                                         00007850
  115 JSWICH=2                                                          00007860
C
C     ALPHA=90-BETA?  ELIMINATE ODD ZONE.                               00007870
C
      IF(ISPLIT.EQ.-1) JSWICH=1                                         00007880
```

```
C
C     NOT IN ZONE 1?  SKIP IT.
C
  120 IF((ISPLIT.EQ.-1).OR.(JCOUNT.NE.0).OR.(DELTA.GT.1E-05)) GO TO 130
      IREF=1
      IF(NSTPTS.NE.1) GO TO 125
C
C     AT END OF AIRFOIL?  ADD FOR TOP AIRFOIL.
C
      IF(X(ILINE,ISWTCH)+DSTSTR.GT.1.0) GO TO 95
C
C     AT END OF ZONE1?  ADD FOR TOP AIRFOIL.
C
      IF(MCOUNT.EQ.0) GO TO 110
      GO TO 93
  125 IF(X(ILINE,ISWTCH)+DSTSTR.GT.1.0) GO TO 128
      IF(MCOUNT.EQ.0) GO TO 130
      IF(MCOUNT-ISPLIT-2)93,95,130
  128 IF(MCOUNT-ISPLIT-1) 95,95,130
  130 CONTINUE
      IF(X(ILINE,ISWTCH)+DSTSTR.LE.1.0) GO TO 140
C
C     COMPUTE VALUES OF C22 AT X= 1.
C
      DELCR=C22R(ILINE,ISWTCH)+(C22R(ILINE,ISWTCH)-C22R(JLINE,JSWTCH))*(
     11.0-X(ILINE,ISWTCH))/DSTSTR
      DELCI=C22I(ILINE,ISWTCH)+(C22I(ILINE,ISWTCH)-C22I(JLINE,JSWTCH))*(
     11.0-X(ILINE,ISWTCH))/DSTSTR
C
C     COMPUTE CHANGE IN LIFT AND MOMENT DUE TO UNDERSHOOT.
C
      DELL1=(C22R(ILINE,ISWTCH)+DELCR)*(1.0-X(ILINE,ISWTCH))/DSTSTR
      DELL2=(C22I(ILINE,ISWTCH)+DELCI)*(1.0-X(ILINE,ISWTCH))/DSTSTR
      DELM1=(C22R(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)+DELCR*(1.0-XSUBO
     1))*(1.0-X(ILINE,ISWTCH))/DSTSTR
      DELM2=(C22I(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)+DELCI*(1.0-XSUBO
     1))*(1.0-X(ILINE,ISWTCH))/DSTSTR
C
C     IF NEEDED ADD FOR TOP AIRFOIL NOT COMPUTED IN HIFOIL.
C
      IF(DELTA.GT.1E-05) GO TO 135
      IF(IREF.EQ.0) GO TO 135
      IF(MCOUNT.GT.(ISPLIT+1)) GO TO 135
      L1=L1-DELL1
      L2=L2-DELL2
      M1=M1-DELM1
      M2=M2-DELM2
  135 L1=L1-DELL1
```

```
      L2=L2-DELL2                                                       00008250
      M1=M1-DELM1                                                       00008260
      M2=M2-DELM2                                                       00008270
      IFIN=1                                                            00008271
  140 CONTINUE                                                          00008280
      RETURN                                                            00008290
      END                                                               00008300


      SUBROUTINE COMPXY                                                 00008310
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00008320
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00008330
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUB0,TNWDST,FMANGL,XLNGTH00008340
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00008350
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M100008360
     3,M2      ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL             00008370
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00008380
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00008390
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00008400
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00008410
C
C     IF FALLS THROUGH,COMPUTE X,Y FOR NEXT ITH POINT ON LINE.(ITH.GT.2)00008420
C
      IF(IHAVEP.EQ.0)GO TO 1                                            00008430
      I=IHAVEP+1                                                        00008440
      X(I,ISWTCH)=X(IHAVEP,ISWTCH)+HDSTRL                               00008450
      Y(I,ISWTCH)=Y(IHAVEP,ISWTCH)-TRNGLH                               00008460
      GO TO 2                                                           00008470
C
C     IF FALLS THROUGH ON 1 X,Y IS ON MACH LINE.                        00008480
C
    1 IF((IATNWL.EQ.1).AND.(JLINE.NE.NGDPTS-1)) GO TO 3                 00008490
      I=IHAVEP+1                                                        00008500
      X(I,ISWTCH)=X(1,JSWTCH)+HDSTRL                                    00008510
      Y(I,ISWTCH)=Y(1,JSWTCH)+TRNGLH                                    00008520
      GO TO 2                                                           00008530
C
C     IF GONE TO 3 COMPUTE FIRST POINT ON RIGHT RUNNING MACH LINE EMANA-00008540
C    1TING FROM THE TUNNEL WALL.                                        00008550
C
    3 I=IHAVEP+1                                                        00008560
      X(I,ISWTCH)=X(1,JSWTCH)+DSTSTR                                    00008570
      Y(I,ISWTCH)=Y(1,JSWTCH)                                           00008580
    2 CONTINUE                                                          00008590
      RETURN                                                            00008600
      END                                                               00008610
```

```
      SUBROUTINE FLUTER                                                 00008620
C
C     FLUTTER COMPUTES THE FLUTTER SPEED AND FREQUENCY.                 00008630
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00008640
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00008650
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH00008660
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00008670
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,L1,L2,M100008680
     3,M2       ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL            00008690
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                          00008700
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                       00008710
      COMMON/BLK2/L3,L4,M3,M4                                           00008720
      COMMON/BLK3/MUU,RSUBA,XSUBA,HAFREQ,OMEGAA,OMEGAH,INCRE            00008730
      COMMON/BLK4/ROOTX1,ROOTX2,ROOTXI,OMEGA,UF,RFREQ,IMAGRT            00008740
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00008750
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00008760
      REAL MUU,INCRE                                                    00008770
      IMAGRT=0                                                          00008771
      RFREQ=2.0/REDFRQ                                                  00008780
      DR=L1*M3-L3*M1-L2*M4+L4*M2                                        00008790
      DI=L1*M4-L4*M1+L2*M3-L3*M2                                        00008800
      CR=MUU*(XSUBA*(M1+L3)-M3+OMEGAA-L1*RSUBA**2-MUU*XSUBA**2)+DR      00008810
      CI=MUU*(XSUBA*(M2+L4)-M4-L2*RSUBA**2)+DI                          00008820
      IF((OMEGAH.LT.1E-06).OR.(OMEGAA.LT.1E-06)) GO TO 50               00008830
      FACTOR=0.5*((L1-MUU)/OMEGAH+(M3-OMEGAA)/OMEGAA)                   00008840
      DISC=FACTOR**2-CR                                                 00008841
      IF(DISC.LE.-1E-06) GO TO 600                                      00008842
      XR1=-FACTOR+SQRT(DISC)                                            00008850
      XR2=-FACTOR-SQRT(DISC)                                            00008860
      GO TO 100                                                         00008870
   50 XR1=CR/(OMEGAA*(MUU-L1)+OMEGAH*(MUU*RSUBA**2-M3))                 00008880
      XR2=XR1                                                           00008890
  100 XI=-CI/(OMEGAA*L2+OMEGAH*M4)                                      00008900
      IF((XR1.LE.-1E-06).OR.(XR2.LE.-1E-06).OR.(XI.LE.-1E-06)) GO TO 60000008901
      ROOTX1=SQRT(XR1)                                                  00008910
      ROOTX2=SQRT(XR2)                                                  00008920
      ROOTXI=SQRT(XI)                                                   00008930
      OMEGA=1.0/ROOTXI                                                  00008940
      UF=OMEGA/REDFRQ                                                   00008950
      WRITE(6,501)RFREQ                                                 00008960
  501 FORMAT(///,5H 1/K=,F20.7)                                         00008970
      WRITE(6,502)ROOTX1,ROOTX2,ROOTXI                                  00008980
  502 FORMAT(///,5X,11H SQRT(XR1)=,E20.7,5X,11H SQRT(XR2)=,E20.7,5X,10H 00008990
     1SQRT(XI)=,E20.7)                                                  00009000
      GO TO 1000                                                        00009001
  600 IMAGRT=1                                                          00009002
      WRITE(6,601)RFREQ                                                 00009003
```

```
  601 FORMAT(///,5H 1/K=,F20.7,///,32H TOO BAD - SQRT(X) IS IMAGINARY.) 00009004
 1000 CONTINUE                                                          00009005
      RETURN                                                            00009010
      END                                                               00009020
```

```
C     PROGRAM B  --  ONE DEGREE OF FREEDOM FLUTTER
C
C     THIS PROGRAM COMPUTES THE FLUTTER FREQUENCY AND FLUTTER VELOCITY
C     OF A CASCADE OSCILLATING AT LOW AMPLITUDES WITH ARBITRARY STAGGER,
C     PHASE LAG, BLADE SPACING,AND FREQUENCY.  THE CASCADE MUST HAVE
C     SUPERSONIC LEADING EDGE LOCUS.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION DATE(3)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2          ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK3/MUU,RSUBA,GSUBA,INCRE
      REAL MUU,INCRE
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I
      COMMON/PCOR/KOUNT
      COMMON/JUNC/IJUNT
      COMMON/STG/STGR
      COMMON/FINE/FIN
      COMMON/IFINE/IFIN,JFIN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
C
C     PRINT NAME OF PROGRAM/DATE OF RUN.
C
      READ(5,7)DATE
    7 FORMAT(3A4)
      WRITE(6,6)DATE
    6 FORMAT(1H1,14(/),14(/),37X,51H OSCILLATING CASCADE FLUTTER PROGRAM
     1   ---RUN OF---,3A4,/,1H1)
C
C     PRINT OUT ALL INPUT INFORMATION VIA INPUT.
C
      CALL INPUT
      DO 1005 JTEM=1,50
      ICO=0
      KOUNT=0
      CALL INTIAL
 1000 CONTINUE
C
C     COMPXY COMPUTES THE VALUE FOR X AND Y GIVEN THE PARAMETERS IHAVEP
C    1AND ISWTCH FOR INDICES (RESPECTIVELY).
```

```
C
      CALL COMPXY
C
C     THE PROGRAM GOES TO 1 IF POINT IS ON INITIAL MACH LINE.
C
      IF((IHAVEP.EQ.0).AND.(IATNWL.EQ.0))GO TO 1
      IF((IHAVEP.EQ.0).AND.(IATNWL.NE.0))GO TO 2
C
C     TEST HERE IF YOU ARE AT A LOW FOIL PT.IF NOT GO TO NEXT POINT.
C
      IF(IATNWL .EQ. 1 .AND. IHAVEP .EQ. KOUNT) GO TO 3
      IF(IHAVEP.EQ.JLINE)GO TO 3
      GO TO 4
C
C     INITIAL POINT(MACH LINE) CALC. FOR U,V,AND C AT 1.
C
    1 CALL MACHLN
      GO TO 5
C
C     HIGH FOIL   CALCULATION HERE FOR U,V,AND C AT 2.
C
    2 CALL HIFOIL
      GO TO 5
C
C     LOW FOIL PT CALCULATION HERE FOR U,V,AND C AT 3.
C
    3 CALL LOFOIL
      GO TO 5
C
C     GENERAL POINT CALCULATION HERE FOR U,V,AND C AT 4.
C
    4 CALL GENFPT
    5 CONTINUE
      IF(IATNWL .EQ. 1 .AND. IHAVEP .EQ. KOUNT) GO TO 25
      IF(IHAVEP.NE.JLINE)GO TO 100
C
C     TEST IF AT END OF FOIL  IF SO COMPUTE Q AND QUIT.
C
   25 IF(X(IHAVEP+1,ISWTCH)+DSTSTR.GT.FIN) GO TO 101
C
C     TEST IF YOU ARE AT A HI AIRFOIL LINE(RIGHT RUNNING MACH) AND IF
C    1SO NOTE THIS BY IATNWL=1 AND JLINE=NGDPTS.
C
      IF(ILINE.EQ.NGDPTS)GO TO 102
C
C     INCREMENT FOR NEXT LINES(FIRST OLD ONE(ILINE) BECOMES LAST NEW 1)
C
      ILINE=ILINE+1
```

```
      JLINE=JLINE+1
      IF(ILINE.EQ.NGDPTS) IATNWL=1
C
C     SWITCH LINES HERE SO FIRST OLD ONE (ILINE) BECOMES LAST NEW ONE.
C    1ZERO OUT NEW LINE POINT INCREMENT COUNTER.
C
  105 IF(ICO.EQ.1) IJUNC=IJUNC-1
      IF(ISWTCH.EQ.1) GO TO 103
      ISWTCH=1
      JSWTCH=2
      IHAVEP=0
      GO TO 1000
  103 ISWTCH=2
      JSWTCH=1
      IHAVEP=0
      GO TO 1000
C
C     AT 102 SET UP FOR HERE ON IN AT TUNNEL WALL.
C
  102 IATNWL=1
      JLINE=NGDPTS
      KOUNT=NGDPTS-1
      GO TO 105
C
C     AT 100 INCREMENT TO NEXT POINT ALONG PRESENT LINE.
C
  100 IHAVEP=IHAVEP+1
C
C     FINISHED TOP AIRFOIL?  TERMINATE
C
      IF((ICO.EQ.1).AND.(IJUNC.EQ.0)) GO TO 106
C
C     PREVENT UNNECESSARY FLOW FIELD CALCULATION.
C
      IF((ICO.EQ.1).AND.(IHAVEP.GT.IJUNC)) GO TO 105
      GO TO 1000
C
C     IN ZONE 1?  TERMINATE
C
  101 IF((JLINE-1).LT.IJUNC) GO TO 106
      ICO=1
      GO TO 105
C
C     FINISH PITCH MODE CALCULATION.
C
  106 FACTOR=DSTSTR/(REDFRQ*REDFRQ)
      M3=M1*FACTOR*2.0
      M4=M2*FACTOR*2.0
```

```
      OMEGAA=MUU*RSUBA**2                                               00001450
      CAPX=1.0-M3/OMEGAA                                                00001460
      DAMP=M4+GSUBA*OMEGAA*CAPX                                         00001470
      RFREQ=2.0/REDFRQ                                                  00001480
      WRITE(6,500)RFREQ,M4,M3,CAPX,DAMP                                 00001490
  500 FORMAT(///,5H 1/K=,F17.7,///,4H M4=,E17.7,5X,3HM3=,E17.7,5X,2HX=,E00001500
     117.7,5X,8HDAMPING=,E17.7)                                         00001510
      IF(JTEM.NE.1) GO TO 210                                           00001520
      IF(DAMP.GT.-1E-06) GO TO 210                                      00001530
      RFREQ=RFREQ-0.5                                                   00001540
      REDFRQ=2.0/RFREQ                                                  00001550
      INCRE=INCRE/5.0                                                   00001560
      GO TO 1005                                                        00001570
  210 IF(ABS(DAMP).LE.1E-06) GO TO 240                                  00001580
      IF(DAMP.GT.1E-06) GO TO 280                                       00001590
      RFREQ=RFREQ-INCRE                                                 00001600
      INCRE=INCRE/5.0                                                   00001610
      IF(INCRE.LE.1E-04) GO TO 240                                      00001620
      GO TO 280                                                         00001630
  240 IF(CAPX.LT.-1E-06) GO TO 1006                                     00001640
      ROOTX=SQRT(CAPX)                                                  00001650
      OMEGA=1.0/ROOTX                                                   00001660
      UF=OMEGA/REDFRQ                                                   00001670
      WRITE(6,501)OMEGA,UF                                              00001680
  501 FORMAT(////,15X,19H FLUTTER FREQUENCY=,E20.7,15X,18H FLUTTER VELOC00001690
     1ITY=,E20.7)                                                       00001700
      WRITE(6,502)JTEM                                                  00001710
  502 FORMAT(////,12H ITERATIONS=,I5)                                   00001720
      GO TO 1007                                                        00001730
  280 RFREQ=RFREQ+INCRE                                                 00001740
      REDFRQ=2.0/RFREQ                                                  00001750
 1005 CONTINUE                                                          00001760
 1006 WRITE(6,503)JTEM                                                  00001770
  503 FORMAT(////,88H TOO BAD - THE FREQUENCY RATIO IS IMAGINARY BUT MAY00001780
     1BE THE CYCLE IS COMPLETE--ITERATIONS=,I5)                         00001790
 1007 CONTINUE                                                          00001800
      STOP                                                              00001810
      END                                                               00001820

      SUBROUTINE INPUT                                                  00001830
C                                                                       00001840
C     SUBROUTINE INPUT READS ALL INPUT.                                 00001850
C     NGRDFN IS THE FINENESS OF GRID NUMBER.                            00001860
C     FSTRMN IS THE FREESTREAM MACH NUMBER.                             00001870
C     RTOSPH IS THE RATIO OF SPECIFIC HEATS.                            00001880
C     REDFRQ IS THE REDUCED DIMENSIONLESS FREQUENCY.                    00001890
C     TNWDST IS THE DISTANCE BETWEEN AIRFOILS.                          00001900
```

```
C     XSUBO IS THE DISTANCE(DIMENSIONLESS) FROM THE LEADING EDGE TO THE
C    1ELASTIC AXIS.
C     STGANG IS THE CASCADE STAGGER ANGLE IN DEGREES.
C     FAZE IS THE UPPER AIRFOIL PHASE LAG IN DEGREES.
C     DELTA IS THE UPPER AIRFOIL PHASE LAG IN RADIANS.
C     MUU IS THE WING DENSITY PARAMETER.
C     RSUBA IS THE RADIUS OF GYRATION.
C     GSUBA IS THE STRUCTURAL DAMPING COEFFICIENT IN TORSION.
C     INCRE IS THE AMOUNT THE REDUCED FREQUENCY IS DECREASED EACH TIME
C    1SQRT(X) IS CALCULATED.
C     OMEGAA IS MUU*RSUBA**2.
C     ROOTX IS THE RATIO OF AFREQ TO THE FREQUENCY OF OSCILLATION.
C     HSUBO & ALPHAO ARE THE MAXIMUM AMPLITUDE(DIMENSIONLESS) OF THE
C    1 AIRFOIL PLUNGING & PITCHING OSCILLATION, RESPECTIVELY.  EACH IS
C    2 SET EQUAL TO 1 SINCE THEIR VALUES ARE INDEPENDENT OF THE COMPUTAT
C     FMANGL IS THE MACH ANGLE.
C     XLNGTH IS THE LENGTH OF THE INITIAL MACH LINE.
C     DELTAS IS THE STEP SIZE OF INCREMENTING ALONG THE MACH LINES.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
C
C     NGDPTS IS THE NUMBER OF GRID POINTS(INCL. XSUBO AND TW POINT).
C     DSTSTR IS THE DISTANCE(HORIZONTAL) OF A STREAMLINE FROM ONE GRID
C    1POINT TO THE NEXT STREAMLINE GRID POINT.
C     HDSTRL IS ONE-HALF OF DSTSTR
C     TRNGLH IS THE HEIGHT OF DELTAS(VERTICAL)
C     U22R IS THE VARIABLE HOLDING THE REAL PART OF U AT THE RIGHTMOST
C    1GRID POINT.THE SECOND INDEX POINTS TO A LINE(MACH) BEING USED.
C     SAME TYPE OF MEANING FOR THE OTHER VARIABLES.
C     AI AND BI ARE USED AS CONSTANTS IN THE DEFINITIONS OF U,V,AND C.
C     COMPUTE ALL VALUESFOR INITIAL MACH LINES INITIAL POINT.
C     SET UP INITIAL INTEGRATION VALUES.
C     ISWTCH IS THE SWITCH VARIABLE FOR CHOOSING THE LINE TO WORK ON.
C     JSWTCH IS THE OPPOSITE OF ISWTCH. IF ISWTCH=1,JSWTCH=2 VICE-VERSA.
C     ILINE CONTAINS THE NUMBER OF POINTS THE LINE EVENTUALLY SHOULD BE.
C     JLINE=ILINE-1 UNLESS AT HI AIRFOILL THEN JLINE=ILINE(JLINE IS TOTA
C    1L NUMBER OF POINTS CONTAINED IN THE OTHER LINE.
C     IHAVE IS A COUNTER TELLING THE NUMBER OF PROCESSED POINTS YOU HAVE
C    1CALCULATED FOR THE LINE YOU ARE WORKING ON.
C     IATNWL=0 SIGNIFIES YOU ARE NOT AT THE HI AIRFOILL YET.IATNWL WILL
C    1BE SET=1 WHEN FINISHED WITH ILINE=NGDPTS(AT HI AIRFOILL).
C
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2        ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
```

```
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK3/MUU,RSUBA,GSUBA,INCRE
      COMMON/JUNC/IJUNT
      COMMON/STG/STGR
      COMMON/FINE/FIN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      REAL MUU,INCRE
      NAMELIST/NAM1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,STGANG,FAZE
     1,MUU,RSUBA,GSUBA,INCRE,FIN
      READ(5,NAM1)
      WRITE(6,NAM1)
      FMANGL=ARSIN(1.0/FSTRMN)
      FNGRDN=NGRDFN
      XLNGTH=TNWDST/SIN(FMANGL)
      NGDPTS=NGRDFN
      FNGDPT=NGDPTS
      DELTAS=XLNGTH/FNGDPT
      TRNGLH=TNWDST/FNGDPT
      HDSTRL=DELTAS*COS(FMANGL)
      DSTSTR=HDSTRL*2.0
      STGR=TNWDST*TAN(STGANG*0.1745329E-01)
      NSTPTS=STGR/DSTSTR
      DTSTSR=NSTPTS*DSTSTR
      IF(STGR-DTSTSR.GE.0.50) NSTPTS=NSTPTS+1
      IF(NGDPTS/2.EQ.NSTPTS-1) NSTPTS=NSTPTS-1
      IF(NGDPTS/2.EQ.NSTPTS+1) NSTPTS=NSTPTS+1
      STGR=NSTPTS*DSTSTR
      STGANG=ATAN(STGR/TNWDST)*57.29578
      IJUNT=NGDPTS/2+NSTPTS
      NSTPTS=NSTPTS+1
      ISPLIT=NGDPTS/2-NSTPTS
      NGDPTS=NGDPTS+1
      S=SQRT(FSTRMN*FSTRMN-1.0)
      DELTA=FAZE*0.1745329E-01
      WRITE(6,5)NGRDFN,FSTRMN,RTOSPH,REDFRQ,TNWDST,XSUBO,STGANG,DSTSTR
     1,MUU,RSUBA,GSUBA,FAZE
    5 FORMAT(////,28H GRID FINENESS INPUT NUMBER=,I10,///,24H FREESTREAM
     1 MACH NUMBER=,F20.7,///,25H RATIO OF SPECIFIC HEATS=,F20.7,///,34H
     2 REDUCED(DIMENSIONLESS) FREQUENCY=,F20.7,///,27H DISTANCE BETWEEN
     3AIRFOILS=,F20.7,///,52H HORIZONTAL POSITION(DIMENSIONLESS) OF ELAS
     4TIC AXIS=,F20.7,///,26H COMPATABLE STAGGER ANGLE=,F20.7,///,9H DEL
     5TA X=,F20.7,///,24H WING DENSITY PARAMETER=,F20.7,///,20H RADIUS O
     6F GYRATION=,F20.7,///,42H TORSIONAL STRUCTURAL DAMPING COEFFICIENT
     7=,F20.7,///,25H UPPER AIRFOIL PHASE LAG=,F20.7)
      WRITE(6,6)
    6 FORMAT(1H1,10X,80HVALUES OF FREQUENCY RATIO FOR VARIOUS NON-DIMENS
     1IONAL FREQUENCIES OF OSCILLATION)
```

```
      RETURN
      END
      SUBROUTINE INTIAL
C
C     INTIAL INITIALIZES ALL FLOW FIELD QUANTITIES.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2        ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I
      COMMON/JUNC/IJUNT
      COMMON/IFINE/IFIN,JFIN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      IJUNC=IJUNT
      AI=.5*REDFRQ*DSTSTR
      BI=.25*REDFRQ*(FSTRMN*FSTRMN/(FSTRMN*FSTRMN-1.0))*DSTSTR
C
C     SET UP INITIAL VALUES FOR U,V,AND C AT (0,0).
C
   50 VRPANL=-1.0/S
      VIPANL=REDFRQ*XSUBO/S
 60   U22R(1,1)=-VRPANL
      U22I(1,1)=-VIPANL
      V22R(1,1)=-U22R(1,1)
      V22I(1,1)=-U22I(1,1)
      C22R(1,1)=-U22R(1,1)
      C22I(1,1)=-U22I(1,1)
      X(1,1)=0.0
      Y(1,1)=0.0
      IF((ISPLIT.NE.-1).AND.(DELTA.LE.1E-05)) GO TO 63
      M1=XSUBO*C22R(1,1)
      M2=XSUBO*C22I(1,1)
      GO TO 64
   63 M1=2.0*XSUBO*C22R(1,1)
      M2=2.0*XSUBO*C22I(1,1)
   64 IF(DELTA.LE.1E-05) GO TO 65
  150 VRPANL=-(COS(DELTA)+REDFRQ*SIN(DELTA)*XSUBO)/S
      VIPANL=-(SIN(DELTA)-REDFRQ*COS(DELTA)*XSUBO)/S
```

```
  160 U33R(1,1)=VRPANL
      U33I(1,1)=VIPANL
      V33R(1,1)=U33R(1,1)
      V33I(1,1)=U33I(1,1)
      C33R(1,1)=-U33R(1,1)
      C33I(1,1)=-U33I(1,1)
      IF(ISPLIT.EQ.-1) GO TO 65
      M1=M1-XSUBO*(C33R(1,1)*COS(DELTA)+C33I(1,1)*SIN(DELTA))
      M2=M2-XSUBO*(C33I(1,1)*COS(DELTA)-C33R(1,1)*SIN(DELTA))
   65 ISWTCH=2
      JSWTCH=1
      ISWICH=1
      JSWICH=1
      IFIN=0
      JFIN=0
      ILINE=2
      JLINE=1
      IHAVEP=0
      IATNWL=0
      JCOUNT=0
      LCOUNT=0
      MCOUNT=1
      RETURN
      END


      SUBROUTINE MACHLN
C
C     MACHLN COMPUTES THE VALUES OF U,V,AND C ALONG THE INITIAL MACH
C    1LINE AT THE GIVEN X VALUE OF THE MACH LINE.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2       ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      ICOUNT=JLINE
      IF(ICOUNT.LT.IJUNC) GO TO 10
      JCOUNT=JCOUNT+1
      KCOUNT=JCOUNT-1
```

```
      IF(KCOUNT.NE.0) GO TO 5                                           00003790
      UR=0.0                                                            00003800
      UI=0.0                                                            00003810
      VR=0.0                                                            00003820
      VI=0.0                                                            00003830
      CR=0.0                                                            00003840
      CI=0.0                                                            00003850
      GO TO 7                                                           00003860
    5 UR=U22R(KCOUNT,3)                                                 00003870
      UI=U22I(KCOUNT,3)                                                 00003880
      VR=V22R(KCOUNT,3)                                                 00003890
      VI=V22I(KCOUNT,3)                                                 00003900
      CR=C22R(KCOUNT,3)                                                 00003910
      CI=C22I(KCOUNT,3)                                                 00003920
    7 K12R=UR+CR+AI*UI                                                  00003930
      K12I=-AI*UR+UI+CI                                                 00003940
      K34R=U22R(1,JSWTCH)-V22R(1,JSWTCH)+BI*(U22I(1,JSWTCH)-C22I(1,JSWTC00003950
     1H))                                                               00003960
      K34I=U22I(1,JSWTCH)-V22I(1,JSWTCH)-BI*(U22R(1,JSWTCH)-C22R(1,JSWTC00003970
     1H))                                                               00003980
      K56R=U22R(JCOUNT,3)+V22R(JCOUNT,3)+BI*(U22I(JCOUNT,3)-C22I(JCOUNT,00003990
     13))                                                               00004000
      K56I=U22I(JCOUNT,3)+V22I(JCOUNT,3)-BI*(U22R(JCOUNT,3)-C22R(JCOUNT,00004010
     13))                                                               00004020
      G1=.5*(K34R+K56R)                                                 00004030
      G2=BI*K12I                                                        00004040
      G3=1.0-AI*BI                                                      00004050
      G4=.5*(K34I+K56I)                                                 00004060
      G5=BI*K12R                                                        00004070
      G6=2.0*BI                                                         00004080
      G7=G3*G3+G6*G6                                                    00004090
      G8=G1-G2                                                          00004100
      G9=G4+G5                                                          00004110
      U22R(1,ISWTCH)=(G8*G3+G9*G6)/G7                                   00004120
      U22I(1,ISWTCH)=(-G8*G6+G9*G3)/G7                                  00004130
      V22R(1,ISWTCH)=.5*(K56R-K34R)                                     00004140
      V22I(1,ISWTCH)=.5*(K56I-K34I)                                     00004150
      C22R(1,ISWTCH)=K12R-U22R(1,ISWTCH)               +U22I(1,ISWTCH)*A00004160
     1I                                                                 00004170
      C22I(1,ISWTCH)=K12I-U22I(1,ISWTCH)               -U22R(1,ISWTCH)*A00004180
     1I                                                                 00004190
      GO TO 503                                                         00004200
   10 W=FSTRMN*FSTRMN                                                   00004210
      T=FSTRMN*FSTRMN-1.0                                               00004220
   20 VRPANL=-1.0/S                                                     00004230
      VIPANL=REDFRQ*XSUBO/S                                             00004240
 30   U=COS(REDFRQ*(W/T)*X(1,ISWTCH))                                   00004250
      V=   SIN(REDFRQ*(W/T)*X(1,ISWTCH))                               00004260
```

```
C
C     HIERARCHY W,T,S,U,V
C
      U22R(1,ISWTCH)=-VRPANL*U-VIPANL*V
      U22I(1,ISWTCH)=-VIPANL*U+VRPANL*V
      V22R(1,ISWTCH)=-U22R(1,ISWTCH)
      V22I(1,ISWTCH)=-U22I(1,ISWTCH)
      C22R(1,ISWTCH)=-U22R(1,ISWTCH)
      C22I(1,ISWTCH)=-U22I(1,ISWTCH)
      IF((ICOUNT.NE.ISPLIT).OR.(DELTA.GT.1E-05)) GO TO 500
      U22R(1,3)=-U22R(1,ISWTCH)
      U22I(1,3)=-U22I(1,ISWTCH)
      V22R(1,3)= V22R(1,ISWTCH)
      V22I(1,3)= V22I(1,ISWTCH)
      C22R(1,3)=-C22R(1,ISWTCH)
      C22I(1,3)=-C22I(1,ISWTCH)
      GO TO 503
  500 IF((DELTA.LE.1E-05).OR.(JLINE.GT.ISPLIT)) GO TO 503
  120 VRPANL=-(COS(DELTA)+REDFRQ*SIN(DELTA)*XSUBO)/S
      VIPANL=-(SIN(DELTA)-REDFRQ*COS(DELTA)*XSUBO)/S
  130 U33R(1,ISWTCH)=VRPANL*U+VIPANL*V
      U33I(1,ISWTCH)=VIPANL*U-VRPANL*V
      V33R(1,ISWTCH)=U33R(1,ISWTCH)
      V33I(1,ISWTCH)=U33I(1,ISWTCH)
      C33R(1,ISWTCH)=-U33R(1,ISWTCH)
      C33I(1,ISWTCH)=-U33I(1,ISWTCH)
      IF(ICOUNT.NE.ISPLIT) GO TO 503
      U22R(1,3)=U33R(1,ISWTCH)
      U22I(1,3)=U33I(1,ISWTCH)
      V22R(1,3)=V33R(1,ISWTCH)
      V22I(1,3)=V33I(1,ISWTCH)
      C22R(1,3)=C33R(1,ISWTCH)
      C22I(1,3)=C33I(1,ISWTCH)
  503 CONTINUE
      RETURN
      END

      SUBROUTINE HIFOIL
C
C     HIFOIL COMPUTES U,V,AND C AT AN UPPER AIRFOIL POINT.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R.U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2         ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
```

```
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/PCOR/KOUNT
      COMMON/STG/STGR
      COMMON/JUNC/IJUNT
      COMMON/IFINE/IFIN,JFIN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      TOPCRD=X(1,ISWTCH)-STGR
      IF(JLINE.EQ.NGDPTS-1) IATNWL=0
      IF(ICOUNT.EQ.0) GO TO 80
      IF(JCOUNT.NE.0) GO TO 75
      K12R=0.0
      K12I=0.0
      J=1
      GO TO 80
   75 K12R=U22R(JCOUNT,3)+C22R(JCOUNT,3)+AI*U22I(JCOUNT,3)
      K12I=-AI*U22R(JCOUNT,3)+U22I(JCOUNT,3)+C22I(JCOUNT,3)
      J=1
   80 IF(ICOUNT.NE.0) GO TO 85
      I=1
      J=2
      K12R=U22R(I     ,JSWTCH)+C22R(I     ,JSWTCH)+AI*U22I(I     ,JSWTCH)
      K12I=-AI*U22R(I     ,JSWTCH)+U22I(I     ,JSWTCH)+C22I(I     ,JSWTCH)
   85 V22R(1,ISWTCH)=-(COS(DELTA)-REDFRQ*SIN(DELTA)*(TOPCRD-XSUBO))/S
      V22I(1,ISWTCH)=-(SIN(DELTA)+REDFRQ*COS(DELTA)*(TOPCRD-XSUBO))/S
90    K56R=V22R(1     ,ISWTCH)
      K56I=V22I(1     ,ISWTCH)
91    K34R=U22R(J     ,JSWTCH)-V22R(J     ,JSWTCH)+BI*(U22I(J     ,JSWTCH)-
     1C22I(J     ,JSWTCH))
      K34I=U22I(J     ,JSWTCH)-V22I(J     ,JSWTCH)-BI*(U22R(J     ,JSWTCH)-
     1C22R(J     ,JSWTCH))
92    G1=1.0-AI*BI
      G2=2.0*BI
      G3=G1*G1+G2*G2
      G4=K56R+K34R-BI*K12I
      G5=K56I+K34I+BI*K12R
      U22R(1     ,ISWTCH)=(G4*G1+G5*G2)/G3
      U22I(1     ,ISWTCH)=(-G4*G2+G5*G1)/G3
      C22R(1     ,ISWTCH)=K12R-U22R(1     ,ISWTCH)+U22I(1     ,ISWTCH)*AI
      C22I(1     ,ISWTCH)=K12I-U22I(1     ,ISWTCH)-U22R(1     ,ISWTCH)*AI
      CPR1=C22R(1,JSWTCH)*COS(DELTA)+C22I(1,JSWTCH)*SIN(DELTA)
      CPI1=C22I(1,JSWTCH)*COS(DELTA)-C22R(1,JSWTCH)*SIN(DELTA)
      CPR2=C22R(1,ISWTCH)*COS(DELTA)+C22I(1,ISWTCH)*SIN(DELTA)
      CPI2=C22I(1,ISWTCH)*COS(DELTA)-C22R(1,ISWTCH)*SIN(DELTA)
      IF(JFIN.EQ.1) GO TO 130
C
C     AT START OF ZONE AND CLOSE TO END?  SKIP LIFT CALCULATION.
```

```
C
      IF((LCOUNT.EQ.0).AND.(TOPCRD+DSTSTR.GT.1.0)) GO TO 130
C
C     AT START OF ZONE OR END OF AIRFOIL?  START/STOP INTEGRATION.
C
      IF((LCOUNT.EQ.0).OR.(TOPCRD+DSTSTR.GT.1.0)) GO TO 95
      M1=M1+2.0*CPR2*(TOPCRD-XSUBO)
      M2=M2+2.0*CPI2*(TOPCRD-XSUBO)
      GO TO 100
   95 M1=M1+CPR2*(TOPCRD-XSUBO)
      M2=M2+CPI2*(TOPCRD-XSUBO)
  100 LCOUNT=LCOUNT+1
C
C     SET UP FOR END OF ZONE CHECK.
C
      IF(ISWICH.NE.1) GO TO 105
      IF(LCOUNT.EQ.IJUNT) GO TO 110
      GO TO 120
  105 IF(LCOUNT.NE.JCOUNT) GO TO 120
C
C     AT END OF ZONE, LINEARLY EXTRAPOLATE LAST VALUE AND STOP INTEGRAL.
C
  110 M1=M1+(2.0*CPR2-CPR1)*(TOPCRD+DSTSTR-XSUBO)
      M2=M2+(2.0*CPI2-CPI1)*(TOPCRD+DSTSTR-XSUBO)
C
C     ZERO COUNTER AND SWITCH ZONE MARKERS.
C
      LCOUNT=0
      IF(ISWICH.EQ.1) GO TO 115
      ISWICH=1
      GO TO 120
  115 ISWICH=2
C
C     ALPHA=90-BETA?  ELIMINATE ODD ZONE.
C
      IF(JCOUNT.EQ.0) ISWICH=1
  120 CONTINUE
      IF(TOPCRD+DSTSTR.LE.1.0) GO TO 130
C
C     COMPUTE VALUES OF C22 AT TOPCRD=1.
C
      DELCR=C22R(1,ISWTCH)+(C22R(1,ISWTCH)-C22R(1,JSWTCH))*(1.0-X(1,ISWT
     1CH)+STGR)/DSTSTR
      DELCI=C22I(1,ISWTCH)+(C22I(1,ISWTCH)-C22I(1,JSWTCH))*(1.0-X(1,ISWT
     1CH)+STGR)/DSTSTR
      DELCPR=DELCR*COS(DELTA)+DELCI*SIN(DELTA)
      DELCPI=DELCI*COS(DELTA)-DELCR*SIN(DELTA)
C
```

```
C     COMPUTE CHANGE IN LIFT AND MOMENT DUE TO UNDERSHOOT.
C
      DELM1=(CPR2*(TOPCRD-XSUBO)+DELCPR*(1.0-XSUBO))*(1.0-TOPCRD)/DSTSTR
      DELM2=(CPI2*(TOPCRD-XSUBO)+DELCPI*(1.0-XSUBO))*(1.0-TOPCRD)/DSTSTR
      M1=M1+DELM1
      M2=M2+DELM2
      JFIN=1
  130 CONTINUE
      ICOUNT=0
      RETURN
      END


      SUBROUTINE GENFPT
C
C     GENFPT COMPUTES U,V,AND C AT A GENERAL FIELD POINT.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2        ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      I=IHAVEP+1
      IF(IATNWL .EQ. 0) GO TO 10
      K12R=U22R(I,JSWTCH)+C22R(I,JSWTCH)+AI*U22I(I,JSWTCH)
      K12I=-AI*U22R(I,JSWTCH)+U22I(I,JSWTCH)+C22I(I,JSWTCH)
      K34R=U22R(I+1,JSWTCH)-V22R(I+1,JSWTCH)+BI*(U22I(I+1,JSWTCH)-C22I(I
     1+1,JSWTCH))
      K34I=U22I(I+1,JSWTCH)-V22I(I+1,JSWTCH)-BI*(U22R(I+1,JSWTCH)-C22R(I
     1+1,JSWTCH))
      GO TO 12
   10 K12R=U22R(IHAVEP,JSWTCH)+C22R(IHAVEP,JSWTCH)+AI*U22I(IHAVEP,JSWTCH
     1)
      K12I=-AI*U22R(IHAVEP,JSWTCH)+U22I(IHAVEP,JSWTCH)+C22I(IHAVEP,JSWTC
     1H)
      K34R=U22R(IHAVEP+1,JSWTCH)-V22R(IHAVEP+1,JSWTCH)+BI*(U22I(IHAVEP+1
     1,JSWTCH)-C22I(IHAVEP+1,JSWTCH))
      K34I=U22I(IHAVEP+1,JSWTCH)-V22I(IHAVEP+1,JSWTCH)-BI*(U22R(IHAVEP+1
     1,JSWTCH)-C22R(IHAVEP+1,JSWTCH))
   12 K56R=U22R(IHAVEP,ISWTCH)+V22R(IHAVEP,ISWTCH)+BI*(U22I(IHAVEP,ISWTC
```

```
     1H)-C22I(IHAVEP,ISWTCH))
      K56I=U22I(IHAVEP,ISWTCH)+V22I(IHAVEP,ISWTCH)-BI*(U22R(IHAVEP,ISWTC
     1H)-C22R(IHAVEP,ISWTCH))
      G1=.5*(K34R+K56R)
      G2=BI*K12I
      G3=1.0-AI*BI
      G4=.5*(K34I+K56I)
      G5=BI*K12R
      G6=2.0*BI
      G7=G3*G3+G6*G6
      G8=G1-G2
      G9=G4+G5
      U22R(I,ISWTCH)=(G8*G3+G9*G6)/G7
      U22I(I,ISWTCH)=(-G8*G6+G9*G3)/G7
      V22R(I,ISWTCH)=.5*(K56R-K34R)
      V22I(I,ISWTCH)=.5*(K56I-K34I)
      C22R(I,ISWTCH)=K12R-U22R(I,ISWTCH)               +U22I(I,ISWTCH)*A
     1I
      C22I(I,ISWTCH)=K12I-U22I(I,ISWTCH)               -U22R(I,ISWTCH)*A
     1I
      IF((ICOUNT.NE.ISPLIT).OR.(DELTA.GT.1E-05)) GO TO 450
      U22R(I,3)=-U22R(I,ISWTCH)
      U22I(I,3)=-U22I(I,ISWTCH)
      V22R(I,3)= V22R(I,ISWTCH)
      V22I(I,3)= V22I(I,ISWTCH)
      C22R(I,3)=-C22R(I,ISWTCH)
      C22I(I,3)=-C22I(I,ISWTCH)
      GO TO 500
  450 IF((DELTA.LE.1E-05).OR.(JLINE.GT.ISPLIT)) GO TO 500
      K12R=U33R(IHAVEP,JSWTCH)+C33R(IHAVEP,JSWTCH)+AI*U33I(IHAVEP,JSWTCH
     1)
      K12I=-AI*U33R(IHAVEP,JSWTCH)+U33I(IHAVEP,JSWTCH)+C33I(IHAVEP,JSWTC
     1H)
      K34R=U33R(IHAVEP,ISWTCH)-V33R(IHAVEP,ISWTCH)+BI*(U33I(IHAVEP,ISWTC
     1H)-C33I(IHAVEP,ISWTCH))
      K34I=U33I(IHAVEP,ISWTCH)-V33I(IHAVEP,ISWTCH)-BI*(U33R(IHAVEP,ISWTC
     1H)-C33R(IHAVEP,ISWTCH))
      K56R=U33R(IHAVEP+1,JSWTCH)+V33R(IHAVEP+1,JSWTCH)+BI*(U33I(IHAVEP+1
     1,JSWTCH)-C33I(IHAVEP+1,JSWTCH))
      K56I=U33I(IHAVEP+1,JSWTCH)+V33I(IHAVEP+1,JSWTCH)-BI*(U33R(IHAVEP+1
     1,JSWTCH)-C33R(IHAVEP+1,JSWTCH))
      G1=.5*(K34R+K56R)
      G2=BI*K12I
      G4=.5*(K34I+K56I)
      G5=BI*K12R
      G8=G1-G2
      G9=G4+G5
      U33R(I,ISWTCH)=(G8*G3+G9*G6)/G7
```

```
      U33I(I,ISWTCH)=(-G8*G6+G9*G3)/G7
      V33R(I,ISWTCH)=0.5*(K56R-K34R)
      V33I(I,ISWTCH)=0.5*(K56I-K34I)
      C33R(I,ISWTCH)=K12R-U33R(I,ISWTCH)+U33I(I,ISWTCH)*AI
      C33I(I,ISWTCH)=K12I-U33I(I,ISWTCH)-U33R(I,ISWTCH)*AI
      IF(JLINE.NE.ISPLIT) GO TO 500
      U22R(I,3)=U33R(I,ISWTCH)
      U22I(I,3)=U33I(I,ISWTCH)
      V22R(I,3)=V33R(I,ISWTCH)
      V22I(I,3)=V33I(I,ISWTCH)
      C22R(I,3)=C33R(I,ISWTCH)
      C22I(I,3)=C33I(I,ISWTCH)
  500 CONTINUE
      RETURN
      END


      SUBROUTINE LOFOIL
C
C     LOFOIL COMPUTES THE VALUES OF U,V,AND C AT A LOWER AIRFOIL POINT.
C
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)
      DIMENSION U33R(200,2),U33I(200,2),V33R(200,2),V33I(200,2),C33R(200
     1,2),C33I(200,2)
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1
     3,M2       ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT
     5,LCOUNT,MCOUNT,ISWICH,JSWICH
      COMMON/BLK5/U33R,U33I,V33R,V33I,C33R,C33I
      COMMON/PCOR/KOUNT
      COMMON/IFINE/IFIN,JFIN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      REAL L1,L2,L3,L4,M1,M2,M3,M4
  999 V22R(ILINE,ISWTCH)=-1.0/SQRT(FSTRMN*FSTRMN-1.0)
      V22I(ILINE,ISWTCH)=(-REDFRQ*(X(IHAVEP+1,ISWTCH)-XSUBO))/SQRT(FSTRM
     1N*FSTRMN-1.0)
   90 K34R=V22R(ILINE,ISWTCH)
      K34I=V22I(ILINE,ISWTCH)
      K12R=U22R(JLINE,JSWTCH)+C22R(JLINE,JSWTCH)+AI*U22I(JLINE,JSWTCH)
      K12I=-AI*U22R(JLINE,JSWTCH)+U22I(JLINE,JSWTCH)+C22I(JLINE,JSWTCH)
      IF (IATNWL .EQ. 0) GO TO 91
      K56R=U22R(KOUNT,ISWTCH) +V22R(KOUNT,ISWTCH)+BI*(U22I(KOUNT,ISWTCH)
     1-C22I(KOUNT,ISWTCH))
      K56I=U22I(KOUNT,ISWTCH)+V22I(KOUNT,ISWTCH)-BI*(U22R(KOUNT,ISWTCH)-
     1C22R(KOUNT,ISWTCH))
```

```
      GO TO 92                                                          00007090
 91   K56R=U22R(JLINE,ISWTCH)+V22R(JLINE,ISWTCH)+BI*(U22I(JLINE,ISWTCH)-00007100
     1C22I(JLINE,ISWTCH))                                               00007110
      K56I=U22I(JLINE,ISWTCH)+V22I(JLINE,ISWTCH)-BI*(U22R(JLINE,ISWTCH)-00007120
     1C22R(JLINE,ISWTCH))                                               00007130
 92   G1=1.0-AI*BI                                                      00007140
      G2=2.0*BI                                                         00007150
      G3=G1*G1+G2*G2                                                    00007160
      G4=K56R-K34R-BI*K12I                                              00007170
      G5=K56I-K34I+BI*K12R                                              00007180
      U22R(ILINE,ISWTCH)=(G4*G1+G5*G2)/G3                               00007190
      U22I(ILINE,ISWTCH)=(-G4*G2+G5*G1)/G3                              00007200
      C22R(ILINE,ISWTCH)=K12R-U22R(ILINE,ISWTCH)+U22I(ILINE,ISWTCH)*AI  00007210
      C22I(ILINE,ISWTCH)=K12I-U22I(ILINE,ISWTCH)-U22R(ILINE,ISWTCH)*AI  00007220
      K=ILINE                                                           00007230
      L=ISWTCH                                                          00007240
      IF((ICOUNT.NE.ISPLIT).OR.(DELTA.GT.1E-05)) GO TO 80               00007250
      U22R(K,3)=-U22R(K,L)                                              00007260
      U22I(K,3)=-U22I(K,L)                                              00007270
      V22R(K,3)= V22R(K,L)                                              00007280
      V22I(K,3)= V22I(K,L)                                              00007290
      C22R(K,3)=-C22R(K,L)                                              00007300
      C22I(K,3)=-C22I(K,L)                                              00007310
      GO TO 94                                                          00007320
   80 IF((DELTA.LE.1E-05).OR.(JLINE.GT.ISPLIT)) GO TO 94                00007330
   84 V33R(K,L)=-(COS(DELTA)-REDFRQ*SIN(DELTA)*(X(K,L)-XSUBO))/S        00007340
      V33I(K,L)=-(SIN(DELTA)+REDFRQ*COS(DELTA)*(X(K,L)-XSUBO))/S        00007350
   86 K12R=U33R(JLINE,JSWTCH)+C33R(JLINE,JSWTCH)+AI*U33I(JLINE,JSWTCH)  00007360
      K12I=U33I(JLINE,JSWTCH)+C33I(JLINE,JSWTCH)-AI*U33R(JLINE,JSWTCH)  00007370
      K34R=U33R(JLINE,L)-V33R(JLINE,L)+BI*(U33I(JLINE,L)-C33I(JLINE,L)) 00007380
      K34I=U33I(JLINE,L)-V33I(JLINE,L)-BI*(U33R(JLINE,L)-C33R(JLINE,L)) 00007390
      K56R=V33R(K,L)                                                    00007400
      K56I=V33I(K,L)                                                    00007410
      G4=K56R+K34R-BI*K12I                                              00007420
      G5=K56I+K34I+BI*K12R                                              00007430
      U33R(K,L)=(G4*G1+G5*G2)/G3                                        00007440
      U33I(K,L)=-(G4*G2-G5*G1)/G3                                       00007450
      C33R(K,L)=K12R-U33R(K,L)+U33I(K,L)*AI                             00007460
      C33I(K,L)=K12I-U33I(K,L)-U33R(K,L)*AI                             00007470
      IF(JLINE.NE.ISPLIT) GO TO 88                                      00007480
      U22R(K,3)=U33R(K,L)                                               00007490
      U22I(K,3)=U33I(K,L)                                               00007500
      V22R(K,3)=V33R(K,L)                                               00007510
      V22I(K,3)=V33I(K,L)                                               00007520
      C22R(K,3)=C33R(K,L)                                               00007530
      C22I(K,3)=C33I(K,L)                                               00007540
   88 CPR2=C33R(K,L)*COS(DELTA)+C33I(K,L)*SIN(DELTA)                    00007550
      CPI2=C33I(K,L)*COS(DELTA)-C33R(K,L)*SIN(DELTA)                    00007560
```

```
      CPR1=C33R(JLINE,JSWTCH)*COS(DELTA)+C33I(JLINE,JSWTCH)*SIN(DELTA)
      CPI1=C33I(JLINE,JSWTCH)*COS(DELTA)-C33R(JLINE,JSWTCH)*SIN(DELTA)
      IF(IFIN.EQ.1) GO TO 140
      M1=M1+2.0*CPR2*(X(K,L)-XSUBO)
      M2=M2+2.0*CPI2*(X(K,L)-XSUBO)
      IF((JLINE.NE.ISPLIT).AND.(X(K,L)+DSTSTR.LE.1.0)) GO TO 94
      M1=M1+(2.0*CPR2-CPR1)*(X(K,L)+DSTSTR-XSUBO)
      M2=M2+(2.0*CPI2-CPI1)*(X(K,L)+DSTSTR-XSUBO)
   94 IREF=0
      IF(IFIN.EQ.1) GO TO 140
C
C     AT START OF ZONE AND CLOSE TO END?  SKIP LIFT CALCULATION.
C
      IF((MCOUNT.EQ.0).AND.(X(ILINE,ISWTCH)+DSTSTR.GT.1.0)) GO TO 140
C
C     AT START OF ZONE OR END OF AIRFOIL?  START/STOP INTEGRATION.
C
      IF((MCOUNT.EQ.0).OR.(X(ILINE,ISWTCH)+DSTSTR.GT.1.0)) GO TO 95
   93 M1=M1-2.0*C22R(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)
      M2=M2-2.0*C22I(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)
      IF(IREF.EQ.1) GO TO 130
      GO TO 100
   95 M1=M1-C22R(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)
      M2=M2-C22I(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)
      IF(IREF.EQ.1) GO TO 130
      IF(X(ILINE,ISWTCH)+DSTSTR.GT.1.0) GO TO 120
  100 MCOUNT=MCOUNT+1
C
C     SET UP FOR END OF ZONE CHECK.
C.
      IF(JSWICH.NE.1) GO TO 105
      IF(MCOUNT.EQ.IJUNC) GO TO 110
      GO TO 120
  105 IF(MCOUNT.NE.(ISPLIT+1)) GO TO 120
C
C     AT END OF ZONE, LINEARLY EXTRAPOLATE LAST VALUE AND STOP INTEGRAL.
C
  110 M1=M1-(2.0*C22R(ILINE,ISWTCH)-C22R(JLINE,JSWTCH))*(X(ILINE,ISWTCH)
     1+DSTSTR-XSUBO)
      M2=M2-(2.0*C22I(ILINE,ISWTCH)-C22I(JLINE,JSWTCH))*(X(ILINE,ISWTCH)
     1+DSTSTR-XSUBO)
C
C     IN ZONE 1?  GET LIFT & MOMENT OF TOP AIRFOIL.
C
      IF(IREF.EQ.1) GO TO 93
C
C     ZERO COUNTER AND SWITCH ZONE MARKERS.
C
```

```
      MCOUNT=0                                                          00008050
      IF(JSWICH.EQ.1) GO TO 115                                         00008060
      JSWICH=1                                                          00008070
      GO TO 120                                                         00008080
  115 JSWICH=2                                                          00008090
C                                                                       00008100
C     ALPHA=90-BETA?  ELIMINATE ODD ZONE.                               00008110
C                                                                       00008120
      IF(ISPLIT.EQ.-1) JSWICH=1                                         00008130
C                                                                       00008140
C     NOT IN ZONE 1?  SKIP IT.                                          00008150
C                                                                       00008160
  120 IF((ISPLIT.EQ.-1).OR.(JCOUNT.NE.0).OR.(DELTA.GT.1E-05)) GO TO 130 00008170
      IREF=1                                                            00008180
      IF(NSTPTS.NE.1) GO TO 125                                         00008190
C                                                                       00008200
C     AT END OF AIRFOIL?  ADD FOR TOP AIRFOIL.                          00008210
C                                                                       00008220
      IF(X(ILINE,ISWTCH)+DSTSTR.GT.1.0) GO TO 95                        00008230
C                                                                       00008240
C     AT END OF ZONE1?  ADD FOR TOP AIRFOIL.                            00008250
C                                                                       00008260
      IF(MCOUNT.EQ.0) GO TO 110                                         00008270
      GO TO 93                                                          00008280
  125 IF(X(ILINE,ISWTCH)+DSTSTR.GT.1.0) GO TO 128                       00008290
      IF(MCOUNT.EQ.0) GO TO 130                                         00008300
      IF(MCOUNT-ISPLIT-2)93,95,130                                      00008310
  128 IF(MCOUNT-ISPLIT-1) 95,95,130                                     00008320
  130 CONTINUE                                                          00008330
      IF(X(ILINE,ISWTCH)+DSTSTR.LE.1.0) GO TO 140                       00008340
C                                                                       00008350
C     COMPUTE VALUES OF C22 AT X= 1.                                    00008360
C                                                                       00008370
      DELCR=C22R(ILINE,ISWTCH)+(C22R(ILINE,ISWTCH)-C22R(JLINE,JSWTCH))*(00008380
     11.0-X(ILINE,ISWTCH))/DSTSTR                                       00008390
      DELCI=C22I(ILINE,ISWTCH)+(C22I(ILINE,ISWTCH)-C22I(JLINE,JSWTCH))*(00008400
     11.0-X(ILINE,ISWTCH))/DSTSTR                                       00008410
C                                                                       00008420
C     COMPUTE CHANGE IN LIFT AND MOMENT DUE TO UNDERSHOOT.              00008430
C                                                                       00008440
      DELM1=(C22R(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)+DELCR*(1.0-XSUBO00008450
     1))*(1.0-X(ILINE,ISWTCH))/DSTSTR                                   00008460
      DELM2=(C22I(ILINE,ISWTCH)*(X(ILINE,ISWTCH)-XSUBO)+DELCI*(1.0-XSUBO00008470
     1))*(1.0-X(ILINE,ISWTCH))/DSTSTR                                   00008480
C                                                                       00008490
C     IF NEEDED ADD FOR TOP AIRFOIL NOT COMPUTED IN HIFOIL.              00008500
C                                                                       00008510
      IF(DELTA.GT.1E-05) GO TO 135                                      00008520
```

```
      IF(IREF.EQ.0) GO TO 135                                           00008530
      IF(MCOUNT.GT.(ISPLIT+1)) GO TO 135                                00008540
      M1=M1-DELM1                                                       00008550
      M2=M2-DELM2                                                       00008560
  135 M1=M1-DELM1                                                       00008570
      M2=M2-DELM2                                                       00008580
      IFIN=1                                                            00008590
  140 CONTINUE                                                          00008600
      RETURN                                                            00008610
      END                                                               00008620


      SUBROUTINE COMPXY                                                 00008630
      DIMENSION U22R(400,3),U22I(400,3),V22R(400,3),V22I(400,3),        00008640
     1C22R(400,3),C22I(400,3),X(400,2),Y(400,2)                         00008650
      COMMON/BLK1/NGRDFN,FSTRMN,RTOSPH,REDFRQ,XSUBO,TNWDST,FMANGL,XLNGTH00008660
     1,DELTAS,NGDPTS,DSTSTR,HDSTRL,TRNGLH,U22R,U22I,V22R,V22I,C22R,C22I,00008670
     2X,Y,S,DELTA,ISWTCH,JSWTCH,ILINE,JLINE,IHAVEP,IATNWL,AI,BI,M1      00008680
     3,M2        ,K12R,K12I,K34R,K34I,K56R,K56I,VRPANL,VIPANL           00008690
     4,STGANG,NSTPTS,ICOUNT,JCOUNT,IJUNC,ISPLIT                         00008700
     5,LCOUNT,MCOUNT,ISWICH,JSWICH                                      00008710
      REAL K12R,K12I,K34R,K34I,K56R,K56I                                00008720
      REAL L1,L2,L3,L4,M1,M2,M3,M4                                      00008730
C                                                                       00008740
C     IF FALLS THROUGH,COMPUTE X,Y FOR NEXT ITH POINT ON LINE.(ITH.GT.2)00008750
C                                                                       00008760
      IF(IHAVEP.EQ.0)GO TO 1                                            00008770
      I=IHAVEP+1                                                        00008780
      X(I,ISWTCH)=X(IHAVEP,ISWTCH)+HDSTRL                               00008790
      Y(I,ISWTCH)=Y(IHAVEP,ISWTCH)-TRNGLH                               00008800
      GO TO 2                                                           00008810
C                                                                       00008820
C     IF FALLS THROUGH ON 1 X,Y IS ON MACH LINE.                        00008830
C                                                                       00008840
    1 IF((IATNWL.EQ.1).AND.(JLINE.NE.NGDPTS-1)) GO TO 3                 00008850
      I=IHAVEP+1                                                        00008860
      X(I,ISWTCH)=X(1,JSWTCH)+HDSTRL                                    00008870
      Y(I,ISWTCH)=Y(1,JSWTCH)+TRNGLH                                    00008880
      GO TO 2                                                           00008890
C                                                                       00008900
C     IF GONE TO 3 COMPUTE FIRST POINT ON RIGHT RUNNING MACH LINE EMANA-00008910
C    1TING FROM THE TUNNEL WALL.                                        00008920
C                                                                       00008930
    3 I=IHAVEP+1                                                        00008940
      X(I,ISWTCH)=X(1,JSWTCH)+DSTSTR                                    00008950
      Y(I,ISWTCH)=Y(1,JSWTCH)                                           00008960
    2 CONTINUE                                                          00008970
      RETURN                                                            00008980
```

00008990

END

LIST OF REFERENCES

Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., Aeroelasticity, Addison-Wesley, 1955.

Drake, D. G., "The Motion of an Oscillating Aerofoil in a Compressible Free Jet," Journal of the Royal Aeronautical Society, v. 60, p. 621-623, September 1956.

Drake, D. G., "The Oscillating Two-Dimensional Airfoil between Porous Walls," Aeronautical Quarterly, v. 8, p. 226-239, August 1957.

Garrick, I. E., "Nonsteady Wing Characteristics," Section F of Aerodynamic Components of Aircraft at High Speeds, Princeton University Press, 1957.

Garrick, I. E., and Rubinow, S. J., Flutter and Oscillating Air-Force Calculations for an Airfoil in Two-Dimensional Supersonic Flow, NACA Report 846, 1946.

Hamamoto, I., "Minute Harmonic Oscillation of Flat Plate Cascades in Supersonic Flow," Undated (probably 1962), in Japanese.

Lane, F., "Supersonic Flow past an Oscillating Cascade with Supersonic Leading-Edge Locus," Journal of the Aeronautical Sciences, v. 24, p. 65-66, January 1957.

Lane, F., "System Mode Shapes in the Flutter of Compressor Blade Rows," Journal of the Aeronautical Sciences, v. 23, p. 54-66, January 1956.

Miles, J. W., "The Compressible Flow past an Oscillating Airfoil in a Wind Tunnel," Journal of the Aeronautical Sciences, v. 23, p. 671-678, July 1956.

Oswatitsch, K., Gas Dynamics, English version by G. Kuerti, Academic Press, 1956.

Platzer, M. F., "Wind Tunnel Interference on Oscillating Airfoils in Low Supersonic Flow," to be published in Acta Mechanica, submitted in November 1971.

Platzer, M. F., and Chalkley, H. G., "Theoretical Investigation of Supersonic Cascade Flutter and Related Interference Problems, Proceedings of the AIAA/ASME/SAE 13th Structures, Structural Dynamics and Materials Conference, San Antonio, Texas, April 1972.

Platzer, M. F., and Pierce, C. V., "Theoretical Investigation of Wind Tunnel Interference Effects upon Fluttering Panels and Airfoils in Low Supersonic Flow," Proceedings of the AIAA/ASME 11th Structures, Structural Dynamics and Materials Conference, Denver, Colorado, April 1970.

Sauer, R., "Elementare Theorie des langsam Schwingenden Uberschallflugels," Zeitschrift fur angewandte Mathematik und Physik, p. 248-253, 1950.

Teipel, I., "Ein Charakteristikenverfahren zur Berechnung der verallgemeinerten ebenen Flatterluftkrafte," Zeitschrift fur Flugwissenschaften, v. 10, p. 374-379, October 1962.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor R. W. Bell, Code 57Be Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Assoc. Professor M. F. Platzer, Code 57Pl Department of Aeronautics Naval Postgraduate School Monterey, California 93940	15
5.	LCDR Henry G. Chalkley, USN Box 344, CFB Summerside Slemon Park, P.E.I. Canada	1
6.	Dr. H. J. Mueller (AIR - 310) Naval Air Systems Command Washington, D. C. 20360	1
7.	Professor M. H. Vavra, Code 57Va Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A Study of Supersonic Cascade Flutter

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*

Aeronautical Engineer's Thesis; June 1972

5. AUTHOR(S) *(First name, middle initial, last name)*

Henry G. Chalkley

6. REPORT DATE June 1972	7a. TOTAL NO. OF PAGES 157	7b. NO. OF REFS 15

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

Supersonic flow past oscillating flat plate cascades with supersonic leading-edge locus is analysed using a linearized method of characteristics valid for arbitrary frequencies and an elementary analytical theory valid only for low frequencies of oscillation. These two methods are extensions of previous work by Teipel and Sauer for the single airfoil in an unbounded supersonic flow to the case of airfoils oscillating in cascade. Included is the determination of pressure distributions and both a two-degree-of-freedom (bending and torsion) flutter analysis and a single-degree-of-freedom (torsion) flutter analysis. Numerically determined flutter boundaries are presented for various primary parameters such as, Mach number, solidity, stagger angle, density ratio, structural damping coefficient, and elastic axis position. Also, results are presented for the related problem of supersonic wind tunnel interference (including the effect of tunnel porosity) and airfoil-airfoil interference.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Cascades						
Flutter						
Supersonic leading-edge locus						
Method of characteristics						

Thesis
C33865
c.1
Chalkley
134927
A study of supersonic
cascade flutter.

134927
Thesis
C33865 Chalkley
c.1
A study of supersonic
cascade flutter.

thesC33865
A study of supersonic cascade flutter.
3 2768 002 09685 1
DUDLEY KNOX LIBRARY